HUMAN PERFORMANCE IN A MULTIPLE-TASK ENVIRONMENT: EFFECTS OF AUTOMATION RELIABILITY ON VISUAL ATTENTION ALLOCATION

A Thesis
Presented to the
Academic Faculty

by

Ralph H. Cullen

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in Psychology

Georgia Institute of Technology

December, 2011

HUMAN PERFORMANCE IN A MULTIPLE-TASK ENVIRONMENT: EFFECTS OF AUTOMATION RELIABILITY ON VISUAL ATTENTION ALLOCATION

Approved by:

Dr. Wendy A. Rogers, Advisor
School of Psychology
Georgia Institute of Technology

Dr. Arthur D. Fisk
School of Psychology
Georgia Institute of Technology

Dr. Francis T. Durso
School of Psychology
Georgia Institute of Technology

Date Approved: August 11, 2011

ACKNOWLEDGEMENTS

I would like to thank Wendy Rogers, Dan Fisk, Frank Durso, Josh Hoffman, and Jerry Duncan for their support and guidance through this process. I would also like to thank all the members of the Human Factors and Aging Lab (especially Chiu Shun Dan) for being around when I needed an extra pair of eyes or ears. Finally (but certainly not least), I'd like to thank my loving fiancée and family for proving the patience and support that made this project as exciting as it was. This research was supported in part by Deere & Company.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Multiple-task environments are pervasive in a variety of workplaces; many jobs require several concurrent, time-sensitive tasks be done in one task space. One concern in these multiple-task environments is attention allocation: To perform well, the operator must be able to know when and where to look. Otherwise, he or she will not be aware of the status of each task or be able to complete them. To aid these jobs, automation has been developed to support attention allocation: Auditory and visual alerts draw attention to where the system determines it is needed. However, imperfect automation may complicate the aid by introducing misses and false alarms to which the operator must also attend.

Researchers studying these environments and automation's purview within them have focused on a variety of different topics. Some examples include: different types of automation (alerts, decision aid systems, etc.), levels of reliability (0-100% reliable), what automation supports (attention allocation to situation awareness to performance), and how automation affects multiple task environments (two tasks to many).

Because attention had not been directly studied in relation to imperfect automation reliability in multiple-task environments, I decided to analyze the effects of different levels of automation reliability on visual attention allocation and how removal of that automation changed those effects. To study this, I helped to develop the Simultaneous Task Environment Platform (STEP), a program to study and test participants' behavior in multiple-task environments. The STEP program enabled me to

vary the frequency and criticality (number of points gained/lost) of the different tasks to disambiguate how automation was affecting the participants.

In the study, participants were trained on all four tasks of the STEP system, had the automation explained to them, and then were asked to gain as many points a trial as possible. There were three between-subject conditions; a system where ~70% of the automated alerts were reliable, one where ~90% of the alerts were reliable, and one where the participants received no automated aid at all. The automation was designed to support visual attention allocation. The participants interacted with the system and automation for twenty-four trials, divided into six blocks over two days, at which point they transferred to a system with no automation at all.

To better understand exactly how the participants interacted with the system, I measured the number of times they accessed each task (attention allocation, as well as a measure of workload) and the number of points they scored (task performance). Mixed ANOVAs for these two measures, as well as a derived measure of efficiency (points scored per window opened), were conducted crossing automation condition with Block (to measure how the participants changed with experience) and task (to measure how certain tasks' attributes affected the way they were acted upon).

Overall, the automation provided a benefit in terms of reduced workload and improved task performance. Participants in the automated conditions opened fewer windows and performed better. This also meant higher efficiency for those conditions. Experience affected conditions differentially. Those in the no automation condition increased their score but also the number of windows opened, causing their efficiency to stay the same. The 70% reliable condition was similar, with a minor point increase and

no significant window decrease, resulting in no significant efficiency gain. The 90% reliable condition gained little score boost, but opened fewer windows by the end of the experiment, becoming more efficient.

The frequency and criticality of tasks affected both the windows opened and the points scored across conditions, as participants in the two automated conditions opened fewer windows and scored relatively more points on those tasks worth many points that did not appear often. This increased their efficiency on those tasks, but also caused them to suffer greater when the automation was taken away.

In the transfer trials, those participants in the automated conditions experienced both a workload increase and a performance decrease. These were centered on the two high-criticality/low-frequency tasks, as the other two tasks showed only small or no change between normal and transfer trials.

These results show that automation at different levels of reliability affects the behavior of the operator of that system differentially based on the attributes of the tasks the operator must oversee. Tasks that happen often and are only important when aggregated over many are not aided by automation as much as those tasks that happen rarely and are critical every time they appear. When automation fails, however, those same tasks that are aided the most suffer the most, whereas those that do not get much aid do not suffer as much. Designers of automated systems should consider the type of tasks to be automated and their attributes, as well as the effects of increasing or decreasing the reliability of the automation when designing automation to provide support to system operators.

CHAPTER 1 – INTRODUCTION

Automated support is ubiquitous in modern environments; it permeates all facets of our lives and we depend on it in many different ways. We use GPS devices to direct us to the correct location. We rely on the fire alarm to warn us of dangerous conditions where we live and work. These automations allow us to take care of other (often many other) tasks while they manage parts of what we need to know. No automation is perfect, however; GPS systems send us down the wrong streets and fire alarms go off when the oven harmlessly overcooks the food. How do these imperfect automations affect the way we go about doing all of the concurrent tasks of which we are in charge?

With my thesis I examined the effects of levels of reliability of an automated aid in a multiple-task system on attention allocation. The automation is a "diagnostic system," providing warning of various system variables at "critical" points. This research informed the results of research in the automation and attention allocation literatures to better explain effects found in the multiple-task literature. Attention allocation had not been studied in a multiple-task environment in the context of automation. Furthermore, this study informs the design of automated systems by helping to explain how imperfect automation affects when and where the operator looks and how that affects performance, both during normal operation and when the automation fails.

Across work domains, people operate systems that provide information from different sources often requiring responses for different tasks. As an added level of difficulty and complexity, successful performance requires tasks to be done concurrently with time constraints on both information presentation and response. These

environments get much of their complexity by nature of being multiple-task environments; more tasks mean more information sources, more task states to keep track of, and more responses required to interact with the environment. Examples of such multiple-task environments are air traffic control, plant operation, and military command and control situations. Other, perhaps less obvious, examples are driving and computer use: the last two of which I will describe as examples.

Driving, even in familiar surroundings, requires execution of many tasks simultaneously and in sequence, including (but not limited to) managing speed, managing position, navigating toward goals, and monitoring for problems. Driving requires the use of many sources of information, such as memory of goals and how to get to a destination, the current position of the controls, the dashboard gauges, and the windshield.

Computer use is another example of an environment with simultaneous multiple tasks: common use requires managing current open tasks, with other tasks occurring simultaneously in other visible or non-visible windows. The inputs and output sources may be physically unchanging (the keyboard and mouse provide inputs and the monitor and speakers outputs) but, depending on the current tasks, the demands and expectations on what happens with those inputs and outputs and when and where to look to efficiently manage time is required to succeed at using the computer.

Challenges of Multiple-Task Environments

Performance requiring attention to concurrent multiple tasks presents increased challenges relative to performance of a single task. Much research has been conducted on attention allocation in dual-task environments (e.g., Navon & Gopher, 1979; Schneider & Fisk, 1982; Wickens, 1980), but only more recently have studies ventured

into more tasks than two (e.g. Elsmore, 1994; Sit & Fisk, 1999). The challenges mentioned here are a subset of those found in the literature, but they represent some that I posit have large effects on a multiple-task, automated environment. Selected challenges include attention activation and inhibition costs, limited cognitive resources, and attention switching costs

Activation Costs

According to the goal-activated model of multiple task environments, tasks attended to are, at the time, the most highly activated goals in the system (Altmann & Trafton, 2002). When another goal is activated, the activation levels of the last goal slowly fade to a lower level. As this level is non-zero, however, new goals must be activated to a level over this resting activation of other goals to which to be attended.

Altmann and Trafton (2002) posited that this activation can happen in one of two ways. Some outside source (such as a light or alarm) can activate a goal over the resting activation level. Or, the goal can be activated by some internal cue telling the user to check a currently unattended goal. This process requires some level of cognitive resources to continue, and may be costly to workload and performance. Therefore, if multiple tasks are occurring simultaneously with no outside task support, then activation costs will be incurred.

Limited Resources

Multiple Resource Theory (MRT; e.g. Wickens, 1984) also gives insights as to why multiple task spaces are inherently more difficult to interact with than single task spaces. Wickens stated that certain characteristics shared across tasks can cause those tasks to interfere with each other. For example, if the primary input for two tasks is

visual, both will suffer from the fact that visual resources are limited and can only process some subset of information at any one time. Tasks in multiple-task environments often share characteristics, so interference is likely.

Switching Costs

Much like activation costs, switching costs are incurred because of the transition from one task to another (Wickens & McCarley, 2008). The difference is that the switching costs are incurred because of the need to switch resources to the new, attended task, not the cost of activating the task itself.

Consistent with MRT, if two tasks are similar in any way, switching between them will incur costs as the new task's information may be confused with the old (Wickens & McCarley, 2008).

Multiplicity of Demands

We can conclude that, by logical inference, multiple tasks cause an overall increase in task demands compared to single tasks, as each task has specific demands that multiply when many tasks are involved.

Human Information Processing and Multiple-Task Environments

To better understand how operators interact with a set of concurrent tasks grouped into a multiple-task environment, I applied the four-stage model of human information processing found in Parasuraman, Sheridan, and Wickens' (2000) study on human interaction with automation to the specific demands in multiple-task environments. The four levels they described are Sensory Processing, Perception/Working Memory, Decision Making, and Response Selection.

Sensory Processing

Parasuraman et al. (2000) defined their first state of information processing, sensory processing, as "the acquisition and registration of multiple sources of information" (p. 287). This refers to knowing where to look and when to look there. Such looking could be based on salient cues from the task itself or self-activation (activation by internal cues to reactivate the task) as suggested in the goal-activation model (Altmann & Trafton, 2002). Sensory processing changes in multiple-task environments, as it includes both receiving and processing the information given by the system (the processing inherent in any system, regardless of task type or number) and, more specifically, knowing which task to attend to when. This can be based on many factors; the perceived importance of the task, the number of times the task is known to happen, and/or some outside influence drawing attention to the task. Sensory processing in multiple task environments is most closely related to the concept of attention allocation, as it is concerned with what information the operator is attending to at any specific time.

Perception/Working Memory

Parasuraman, et al. (2000) named this class of functions "information analysis": "conscious perception…manipulation of processed and retrieved information in working memory…[and] rehearsal, integration, and inference" (p.287). In other words, it refers to the operator's comprehension of the current state of the overall task space. In a multiple-task environment, this involves two planes of abstraction, the current understanding of the task space as a whole and the understanding of each task's individual state. This stage is linked to the concept of Situation Awareness (SA), and thus can be defined at

different levels, from a basic understanding of the parameters given by the system itself (level one SA), to a general understanding of the state of the task (level two SA), to the ability to predict future states of the task (level three SA) (Endsley, 1995).

Decision Making and Response Selection

The last stages defined by Parasuraman, et al. (2000) referred to knowing what to do and when to do it and then executing that action correctly. Decision Making was defined as, "where decisions are reached based on [processing in earlier stages]" (p.287), whereas Response Selection was, "the implementation of a response of action consistent with the decision choice" (p.287). In multiple-task environments, the multiplicity of tasks increases these demands. The current study, however, is interested in automation that supports earlier stages.

Sensory Processing in an Automated Multiple-Task Environment

The importance of the sensory processing stage in a multiple-task environment, as stated earlier, is that it transcends just the intake of information; it deals with attention allocation between tasks. Because further processing and responding to tasks requires them to first be found, understanding the attention allocation between tasks in multiple task environments is a crucial first step in gaining a better understanding of operator interactions. The goal in the present study was to focus on sensory processing (hereafter operationalized as attention allocation) in the context of automation, one possible aid posited in the literature to alleviate some of the workload required to manage a multiple-task environment (Bliss & Acton, 2003; Ma & Kaber, 2007).

Automation and Possible Effects on Attention Allocation

Automation can support attention allocation by directing the operator's attention to critical or important tasks at the time they become important, thereby alleviating the need to search each task to determine which need to be acted upon. Common automations of this type include warning alarms and lights on consoles as well as flashing windows and auditory alerts on computers. Diagnostic automations, those that aid attention allocation by alerting the operator where to look, with reliability higher than ~70% were shown to have a higher benefit than cost for operators using them (Wickens & Dixon, 2007).

Attention Allocation and Automation Reliability

The alarm and alert literature is often concerned with how attention is affected when an alert or alarm happens and the converse, how attention affects alarm response. Low reliability levels elicit an effect called the "cry wolf syndrome" in operators (Breznitz, 1984). The cry wolf syndrome is a lack of response to alarms that have been deemed to be unreliable. This causes the alarms not to be attended to. Additionally, high levels of workload negatively affect the ability of the operator to respond to alarms, limiting the amount of attention given to them (Bliss & Dunn, 2000).

Researchers not specifically measuring attention allocation have noticed that automations with low reliability have less of an effect on where the participant acts, not affecting their attention allocation as often (Bliss & Acton, 2003). They also engender more double-checking of the task governed by the automation in between automation alert periods (Ma & Kaber, 2007). Highly reliable alerts, however, are followed more often; the participants allocated their attention with the alert.

Attention Allocation, Automation Reliability, and the Matching Law

Herrnstein (1961) stated that in situations where multiple tasks are placed on concurrent, variable interval schedules, the rate of responding to an option matches, in a linear relationship, the rate of reinforcement of that option. He called this the matching law. In the current context of automation and multiple-task environments, the matching law could be applicable in two ways.

First, if the options are the tasks and the reinforcement is the feedback of a correct answer, the purest form of the matching law would dictate that those tasks that occurred more frequently would be the ones that were responded to the most, as they provided the more frequent reinforcement.

Differences between the tasks and the discriminability of the different schedules, however, are not accounted for in pure matching law. Baum, in 1974, posited two phenomena to account for these differences: undermatching (a lack of a perfect matching relationship between choices) and bias (the preference of one option of another above and beyond frequency). In an environment with multiple different tasks on quick schedules, the problem of discriminating task schedules becomes more difficult, as more workload is placed on doing the tasks themselves. This, Baum (1974) suggested, causes undermatching. Furthermore, the cognitive differences between task types, as well as the quality of reinforcement (i.e., the number of points gained and lost), can cause bias towards certain tasks away from others. An understanding of these different factors would help explain the allocation strategies of users of a multiple-task system.

Second, if the opportunities to choose a different task to attend to are perceived as the options and the tasks needing a response (at a critical state) as the reinforcement, the

matching law could inform about the reliance on the automation. Under the matching law, better automation (automation with higher reliability) would be responded to (followed) more often, as it would provide a more efficient way to match the reinforcement (the task at a critical state) to the response (the opening of the window). This might also explain how some tasks might be attended to more; the more feedback the task provides, the more reinforcing it is to visit that task.

Summary

With the added variable of diagnostic automation, studying attention allocation in a multiple-task environment becomes all the more important. Automation has been posited to help with the workload induced by multiple tasks, but mixed effects have been reported in the literature, with the reliability of the automation affecting exactly how helpful (or hurtful) the automation is to operator performance. In this study, I was interested in diagnostic automation, what it is *intended* to affect (attention allocation), and what was *actually* affected by different levels of automation reliability.

Overview of This Study

Identified Gaps in the Literature

Although much research has been focused on automations that aid attention allocation (the alert and alarm literature fall under this category), the effects of such automated aids had not been researched in a multiple-task environment where the efficient allocation of visual attention may be very important.

Focus of This Research

The purpose of the research was to ascertain whether automation reliability in a multiple task environment affects attention allocation and, if so, how. I focused

especially on attention allocation as few studies have measured it explicitly in multiple task environments. This focus was accomplished by making the automated aids diagnostic to support attention allocation.

To inform this question, participants controlled a multiple-task environment. The level of automation reliability varied across participants (~70%, ~90%, or no automated aid).

The levels of reliability were selected on the basis of the literature. In Wickens and Dixon's (2007) meta-analysis, they estimated the level at which the reliability of an automation neither harms nor helps the performance of the participant at 70%, with a confidence interval between 63% and 77%. The present 70% level was created to reflect this crossover point. The 90% level was created to be well above this crossover point but not perfect; that is, it would purportedly provide some performance benefit to the participant, but still make mistakes.

Sit and Fisk (1999) found that the allocation strategies between tasks that participants select differ for each participant. To combat this, I used a point structure to determine the relative importance of each of the tasks, as well as devised timing schedules between tasks that lend themselves to specific checking strategies. Setting up a specific point and timing structure allows me to create optimum allocation strategies and better understand and interpret the results.

I predicted that, without automation, participants would balance their checking across the four tasks, as they were explicitly told that each task would reward them with the same number of points over a trial. Their tendency to match their responses to the tasks that happen the most frequently (Herrnstein, 1961) would be balanced out by the

above explicit instructions and their bias toward not missing the higher point tasks (Baum, 1974). However, participants in the automated conditions would be expected to, at some level, follow the alerts given by the automation (Bliss & Acton, 2003). The more reliable the automation, the more the participants should allocate their attention to where it indicates (Herrnstein, 1961).

Also of interest in this study was the effect of a total automation failure. Previous studies showed that a loss of automation affected users differentially depending on the level of automation originally provided (Ma & Kaber, 2007). At the end of six blocks of trials, I removed the automation entirely. This was done to investigate whether previous experience with different levels of automation reliability affects how users respond to a total automation failure.

Working to answer these questions will support both the further study of multiple-task environments and the design and function of automations used to aid operators. A better understanding of the effects of automation reliability at the beginning of the flow of operator completion of the task (attention allocation) might better explain why certain effects are found in relation to the later stages of the flow (awareness and performance). This, in turn, will aid design by supporting the creation of automation that directs attention to the most critical parts of the task when and where it is needed as well as designing for automation failures.

Simultaneous Task Environment Platform (STEP) Program

I have developed an experimental platform designed to mimic the cognitive demands of a prototype multiple-task environment (see Cullen, Dan, Arivazhagan, & Rogers, 2011 for the full specifications of the program). The basis and inspiration for

this program was taken from SYNWORK1 (Elsmore, 1994; Sit & Fisk, 1999). This new platform is designated the Simultaneous Task Environment Platform (STEP). STEP provides the ability to vary the tasks and task parameters as well as the actions of the automation to assess the questions set forth by the present research.

The tasks and configuration of STEP in this study were developed to be representative of the cognitive demands of work. The four tasks chosen (memory search, visual search, reset, and event response) each represent a cognitive construct used in operational environments, as described below in the context of driving.

The memory search task simulates short-term retrieval of information provided by the environment; much like having to remember a set of directions and recall them in series while driving. The visual search task simulates tasks that require the operator to pick out a matching stimulus from a set of distracters, such as turning down the right street. The reset task simulates the task of responding to a measurable critical event; that is, responding when an event becomes critical while being able to track its progress toward criticality. An example of this is shifting a manual vehicle by watching the tachometer; a perfect shift occurs when the tachometer is in the exact right place, not before or after. Finally, the event response task represents tasks where an unpredictable event happens and must be responded to in a certain amount of time, like a critical warning light appearing on the dashboard.

To simulate the limits of attention, the four tasks were obscured by a window when not viewed; to view a task, the participant had to click on that task's window to open it. Opening one task closed the last one. This allowed the number of views and the relative amount of time spent on each task to be successfully and easily measured. This

simulates the fact that when attending to one task, it is often not possible to simultaneously attend to another task.

CHAPTER 2 – METHOD

Participants

There were total of 60 participants, 20 for each of the conditions: two levels of automation reliability and a no-automation control condition. There were 25 females and 35 males (see Table 1 for participant details). The participants were students of the Georgia Institute of Technology, recruited through Experimetrix, an online system designed to allow students to receive extra credit in psychology courses by participating in experiments. All participants were screened to have visual acuity of 20/40 or better using Snellen near and far eye charts (Snellen, 1968).

Materials

The experiment used the STEP program (detailed below; see also Cullen at al., 2011) installed onto Windows-based PCs. The ability assessments were completed or recorded using pen and paper.

Demographics and Health Questionnaire

A demographic and health questionnaire was completed by all participants (Czaja et al., 2006) to make sure the sample collected was representative. Demographic information such as age was collected from this questionnaire (see Table 1).

Ability Tests

To assess visual acuity, perceptual speed, memory span, and vocabulary, respectively, the Snellen Eye chart (Snellen 1868), Digit Symbol Substitution (Wechsler, 1997), Reverse Digit Span (Wechsler, 1997), and the Shipley Vocabulary (Shipley, 1986) tests were administered. These tests were used to assess whether participants in the

different conditions varied in abilities. Table 1 illustrates that no significant differences were found across conditions for age or the ability measures.

Table 1. Summary of age and abilities test data.

	No Automation		**70% Reliability**		**90% Reliability**		**ANOVA**	
	Mean	SD	Mean	SD	Mean	SD	F	p
Age	20.42	2.43	19.75	1.21	19.95	1.10	0.82	0.45
Shipley Vocabulary	31.60	3.12	31.00	3.69	30.85	3.17	0.28	0.75
Reverse Digit Span	9.75	2.36	10.55	2.09	10.95	2.54	1.37	0.26
Digit Substitution	69.20	9.67	68.60	15.53	71.40	7.04	0.34	0.71

STEP Program

The STEP program is an experimental platform designed to represent the task demands of multiple-task environments (see Cullen, Dan, Arivazhagan, & Rogers, 2011 for the full specifications of the program). The current study used the following tasks and configuration: The task space was divided in half on both axes into four identical quadrants, top-left, top-right, bottom-left, and bottom-right. Each quadrant housed a separate, independent task. In the center of the screen was a small box denoting the current overall score (summed across all four tasks). A sample layout showing the quadrants, the tasks in each, and the center score box is shown in Figure 1. It is important to note that this is only an explanatory diagram; the actual experiment layout was different, based on the windows explained later and certain task parameters (e.g., the memory set would have never been shown at the same time as the target letter). The four tasks were, from top-left to bottom-right, the memory search task, the visual search task, the reset task, and the event response task.

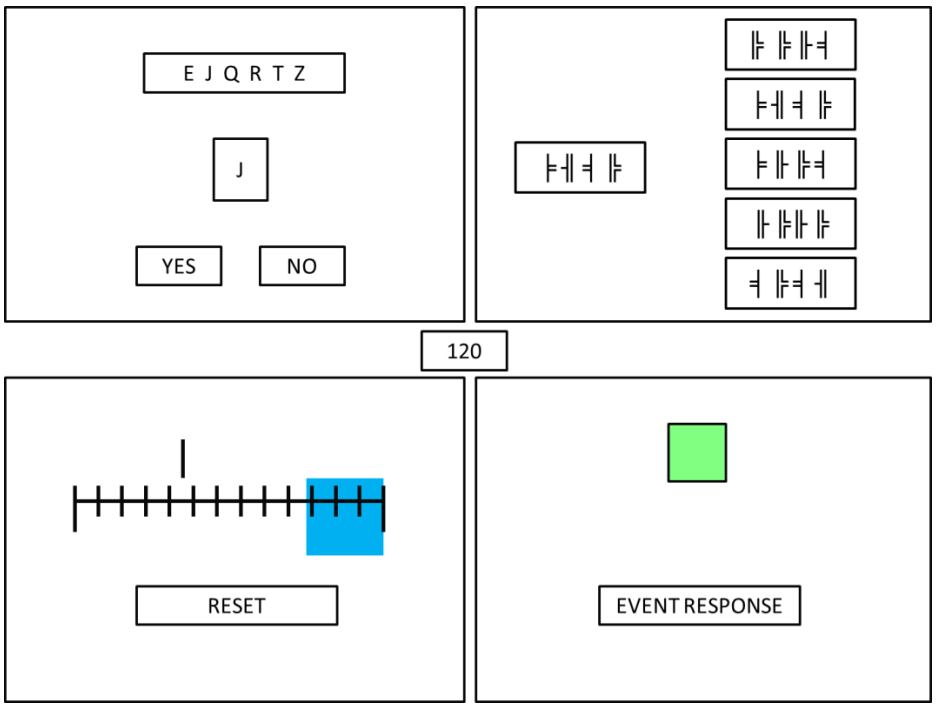

Figure 1. The task layout of the STEP program in the current study. The four tasks were, from top-left to bottom-right, the memory search task, the visual search task, the reset task, and the event response task.

The memory search task, placed in the top left, was a modified Sternberg memory search task, adapted from Sternberg (1969) and Fisk and Rogers (1991). The participants were given a set of letters to remember, the set was taken away, a target stimulus letter was presented, and the participants had to respond within 10 seconds as to whether the target stimulus was part of the set they were presented. If they were able to correctly determine which set the letter came from, they added points to the score, whereas no answer or an incorrect one took them away. This task, as well as the specific point structure, is discussed in more detail in Appendix A.

The visual search task, placed in the top right, had one target barcode on the left and five barcodes on the right. The participants were tasked with trying to find the barcode on the right that matched the one on the left before the next barcode set appeared 10-20 seconds later. Points were gained for choosing the right barcode and lost for choosing the wrong barcode or letting the task timeout. This task, as well as the specific point structure, is discussed in more detail in Appendix B.

In the reset task, placed in the bottom left, the indicator bar started on the left and moved toward the right. Pressing the reset button awarded points to the participant based on how far the bar was from the left; more points being awarded the closer the bar was to the highlighted area. Maximum points were awarded for resetting the bar when in the highlighted area. If the bar was allowed to reach the end, however, the bar reset to the left end and points were lost. This task, as well as the specific point structure, is discussed in more detail in Appendix C.

The event response task, in the bottom right, featured a box and a button. The box started with a white fill and stayed that way, requiring no response. At some points throughout each trial, however, the box changed to green, denoting that an event had occurred. When the box changed, the participants had 10 seconds to respond before the box returned to white. They gained points by responding to the green box in time. They lost points by not responding to the green box or by responding to the white box. This task, as well as the specific point structure, is discussed in more detail in Appendix D.

System Strategy Development

Because of the aforementioned strategy concerns pertaining to Sit and Fisk (1999), I developed a set of timing and point reward rules to disambiguate attention

allocation in the automated and non-automated conditions. Each task rewarded the same number of points for overall correct performance and deducted the same number of points for overall incorrect performance over each trial. This does not mean each task rewarded the same number of points for a correct response, but that, across the entire trial, the same number of points were possible for performing each trial correctly. The participants were told this to make sure that they had no explicit reason to favor one task over another.

To work toward the goal of disambiguating attention allocation strategies, I varied two separate task attributes: frequency and criticality. Frequency was defined as the number of times per trial the task required a point-dependent response (a response that gained or lost the participant some amount of points). High-frequency tasks happened 20 times per trial, whereas low-frequency tasks happened 5 times per trial. Criticality was the number of points gained or lost by responding to the task. High-criticality tasks awarded 120 points every time they were done correctly and deducted 60 points every time done incorrectly. Low-criticality tasks awarded 30 points every time they were done correctly and deducted 15 points every time done incorrectly. These two task attributes were combined to make two types of tasks, high-frequency/low-criticality and low-frequency/high-criticality. Table 2 shows these two task types graphically, as well as which tasks in the STEP platform fell into each task type.

The tasks were allocated to each task type based on certain attributes of the tasks themselves. For example, the memory search and event response tasks were chosen to be the low-frequency tasks due to the fact that they would have been too easy to complete if

they happened too often. The other two tasks would not change in difficulty regardless of the number of times they happened per trial, so they became the high-frequency tasks.

Table 2. The two task attributes and how each task is categorized.

	HIGH CRITICALITY **Gain 120 Points** **Lose 60 Points**	**LOW CRITICALITY** **Gain 30 Points** **Lose 15 Points**
HIGH FREQUENCY **(20 times per trial)**		Visual Search Task Reset Task
LOW FREQUENCY **(5 times per trial)**	Memory Search Task Event Response Task	

It is important to note that not all possible tasks are represented here; high-frequency/high-criticality tasks (and their counterpart, low-frequency/low-criticality tasks) are possible in work environments, but performance in those tasks would not be expected to differ between conditions. Important tasks that happen a lot would likely be attended to often, whereas unimportant task that occur infrequently would likely not be attended to very much.

Windows

To simulate the dynamic nature of shifting between tasks in work environments, each task was hidden by a window (Figure 2). Windows could be opened one at a time with the opening of one window causing the closing of the last. Thus only one task was visible at a time, which provided an index of where a participant's attention was at any one time.

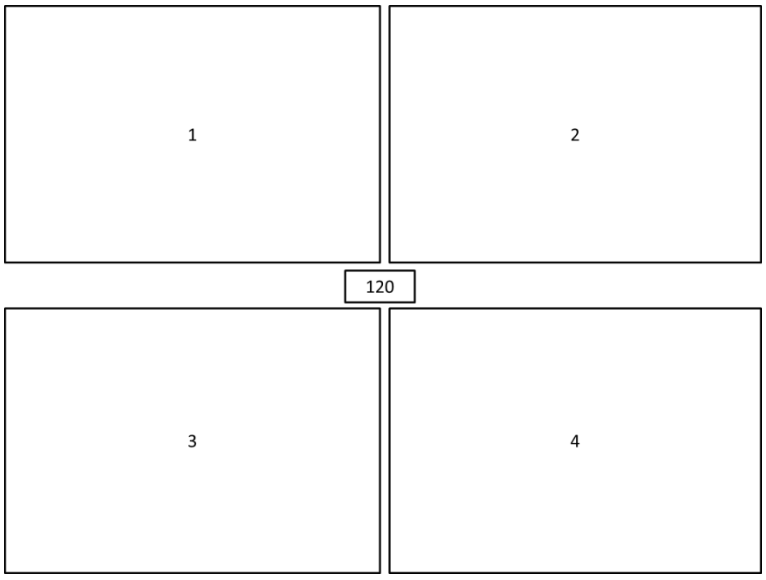

Figure 2. The task layout, including the windows obscuring the four tasks.

During piloting, I discovered that participants ignored the automation and instead adopted a strategy of opening as many windows as possible. Their performance was also at ceiling, as they would be able to see every task and respond to it in time. To better simulate work tasks, an explicit efficiency cost was placed into the task: each window opened would cost the participant two points. Further piloting determined that, this successfully lowered the participants' performance level equally for all four tasks.

Automated Aids

The automated aids in the experiment were red borders around tasks that the automation determined to be at a "critical" state, with each task having certain states determined to be "critical". An example is shown in Figure 3. The critical states for the four tasks were as follows: the appearance of the test stimulus letter in the memory search task; the barcodes appearing in the visual search task; the indicator bar beginning to move in the reset task; and the box changing color in the event response task.

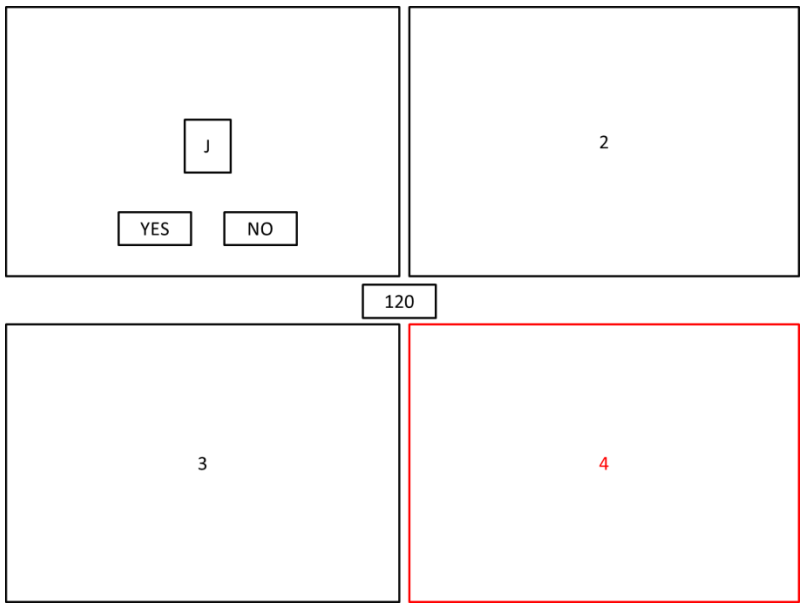

Figure 3. The task layout as it might look in the experiment. Note the open window in the top left showing the memory search task and the automated warning in the bottom right for the event response task.

The automations for each task were all set to the level of reliability determined by the participant's condition (i.e., high or low). Automation failures were evenly divided between misses (no red box when critical state is reached) and false alarms (a red box when no critical state is reached).

Reliability Calculation

The reliability of each condition was created to be approximately 70% and 90%. Because of the need to counterbalance across conditions, a set number of task events were defined within each block (200). Across four trials (one block), all three conditions (70%, 90%, and no automation) had the same number of tasks and the same maximum possible score. The reliability calculation was complicated by the fact that, when a miss was added, it had to replace a hit (because of the set number of task events), but the false

alarms had to be added to the total beyond the task events. For example, if one miss and one false alarm were added, there would be 199 hits, 1 miss (using the 200th task event), and 1 false alarm, totaling 201.

Reliability was calculated by dividing the number of hits by the total number of hits, misses, and false alarms (all possible system actions). Because it was important to make sure each task had similar percentages of hits and misses, the two high incidence tasks had a failure rate of four times the amount of the low incidence ones.

For the 90% condition, the system actions were distributed as follows: 190 hits, 10 misses, and 10 false alarms. The high incidence tasks had 4 false alarms and 4 misses each, whereas the low incidence tasks had only one false alarm and one miss each. The reliability was calculated to be 90.48%, as shown in Table 3.

For the 70% condition, the events were distributed as follows: 160 hits, 40 misses, and 40 false alarms. The high incidence tasks had 16 false alarms and 16 misses each, whereas the low incidence tasks had 4 false alarms and 4 misses each. The reliability was calculated to be 66.67%, as shown in Table 3.

Table 3. The distribution of hits, misses, and false alarms in each automation condition.

	70% Condition	**90% Condition**
Hits	160	190
Misses	40	10
FAs	40	10
Reliability	$\frac{(160\ Hits)}{(160\ hits + 40\ misses + 40\ FAs)} = 66.67\%$	$\frac{(190\ hits)}{(190\ hits + 10\ misses + 10\ FAs)} = 90.48\%$

Event Definition

A task event was defined as the point at which the task requires a point-dependent response from the user. An automation event was defined as when it appears, attempting to attract the user's attention. A "hit" was defined by being both a task event (something happens in the task) and an automation event (the automation warns of something happening in the task). A "miss" was a task event with no automation event. A "false alarm" was an automation event with no task event. A "correct rejection" was a lack of any event, task or automation.

Timing Schedules and Counterbalancing

To avoid several system actions initiating at the same time, all events (task and automation) were placed on timing schedules. Because there were exactly 50 task events per trial and exactly 300 seconds per trial, it was determined that an event should occur, on average, every 6 seconds. To randomize the sequence, timing schedules were created by taking 50 random integer values between four and eight (averaging at six), adding them sequentially to create a set of numbers increasing from 0 to 300, and placing an event at each value. This made each task event (hit or miss) occur between 4 and 8 seconds after the preceding task event. The task placed at each value was determined using the following rules: Tasks two and three (the high incidence tasks) occurred once every 10-20 seconds. Tasks one and four (the low incidence tasks) occurred once every 50-70 seconds. The starting task was randomly selected. At the end of this process, the resulting timing schedules behaved as four concurrent variable-interval schedules. Four such timing schedules were created. Participants in different conditions received the

same task event schedules. The only thing changed between conditions was the number of false alarm and miss events.

The incidence of misses was determined by randomly selecting certain hit events and removing the associated automation event, leaving only the task event. The incidence of false alarms was determined by randomly selecting a number between 0 and 300, and creating an automation event at that second in the trial with no associated task event.

Procedure

Participants completed the study across two days (see Appendix E for details). Participants first read and filled out an informed consent form (shown in Appendix F). Upon providing informed consent, participants completed the demographic and health questionnaires. The participants were then given a short break.

Each task was then presented individually to the participant in its specific quadrant. The participants had the task explained to them through written and vocal instructions, then practiced each task individually ten times. A short break followed.

After each of the four tasks was trained in isolation, they were trained as a whole, adding the windows and any automation for one five minute training session. A short break followed.

There were six blocks of four trials. All six blocks used the same four timing schedules, but the timing schedules were randomly distributed within the blocks. The six blocks combined everything learned in training: the four tasks ran concurrently with the windows and prescribed automation condition. The first three blocks of trials took place on the first day of testing, whereas the second three were on the second day of testing,

conducted approximately 48 hours after the first. Each trial lasted 5 minutes, long enough for the participant to develop and keep a strategy, but short enough so that the participant did not become fatigued. The participants were offered a break after every trial and required to take a 2 to 5 minute break after every block. At the end of the first day (the first three blocks), the participants took the Snellen eye chart (Snellen, 1868), and Shipley Vocabulary (Shipley, 1986) ability tests and then briefed on when they next needed to show up to the experiment.

At the beginning of the second day, the participants were briefly reminded of the goal of participation and the structure of the system. There were given one 5 minute refresher trial to remind them of the four tasks and overall system. They then completed the final three blocks of experimental trials. After the sixth block of trials, all participants took the Reverse Digit Span and Digit Symbol Substitution (Wechsler, 1997) ability tests.

Participants were then asked to complete one more set of four 5 minute trials. This was the transfer block, where all of the participants were transferred to a system with no automated aid. The participants in the no automation condition were told of no changes as they were using the same system as before. The participants in the automated conditions were told that they would have no automated aid. After all the transfer trials were finished, the participants were asked to complete a strategy questionnaire (presented in Appendix G). The purpose of this questionnaire was to assess what the participants thought of the automated aids and for them to record any strategies they had used to find critical tasks. The results of this strategy questionnaire will not be presented here. They were then debriefed (shown in Appendix H) and compensated for their time.

Design

There were three independent variables in this study, one between-participants variable (automation condition) and two within-participants variables (task and block). Automation condition was the level of automation reliability given to the participants (no automation, 70% reliable automation, 90% reliable automation). Task refers to the four different tasks (memory search, visual search, reset, and event response). The data from the individual trials were combined into blocks, with the data for each block being an average of four trials. Learning, then, was measured across blocks, starting at block 1, the first set of experimental trials on the first day, and continuing ultimately to block 6, the last set of experimental trials on the second.

There were three primary dependent measures. Attention allocation was measured every trial as the number of accesses of each task (taken from the number of times the task's window is opened) and for the trial overall. Score was measured every trial as the points the participant earned for each task and for the trial overall. Based on these two measures, a third measure was created: efficiency, defined as the number of points gained per window opened. Efficiency was calculated for each task in a trial and for the trial overall.

CHAPTER 3 – RESULTS

For each dependent variable (windows opened, points scored, and efficiency), I conducted a three-way mixed ANOVA crossing automation condition (none, 70% reliable, or 90% reliable), task (memory search, visual search, reset, and event response), and block (blocks 1-6).

I made the decision to analyze at the block level for several reasons: First, I conducted the ANOVA at each level of the time variable (trial by trial, block by block, and day by day) and few things differed. Second, the reliability level was calculated and normalized at the block level; the small amount of events per trial made equalizing each trial impossible. Third, the transfer block at the end of the study was made up of four trials just like the experimental blocks, so it was best to compare across a similar number of trials. The analyses at the trial and day levels, as well as the comparable data from the block level, can be found in Appendix I.

Windows Opened

The measure of windows opened was created to inform two different things. One, the overall number of windows opened across a trial provided a measure of workload; the more windows opened, the more workload put into switching visual attention between tasks (Wickens & McCarley, 2008). Second, the relative number of windows opened across tasks provided insight into the allocation strategies of participants, in terms of which windows they opened most.

Opening windows resulted in points lost and presumably increased workload; therefore, the fewer windows the participants opened, the better (to a point). Given that

the participants needed to open a certain number of windows to complete all the tasks and attain maximum points, the optimum number of windows to open across a trial would be the minimum needed to complete each task. This worked out to 5 windows each for the high-criticality/low-frequency tasks and 20 each for the low-criticality/high-frequency tasks. The optimum number of windows for a trial, then, was 50 (20 * 2 + 5 * 2).

The ANOVA results are presented in Table 4. All of the effects and interactions were significant at the $\alpha = .05$ level. Power for all effects was above .99. The analysis of the two within-participants variables violated Mauchly's sphericity test ($p < 0.001$), so the Greenhouse-Geisser correction was used in all cases where the task and block variables were involved. All follow-up pairwise comparisons were Bonferroni corrected at the level of $\alpha = .05/$(the number of analyses done). Each effect is discussed in more detail in the following sections.

Table 4. Three-way mixed ANOVA for the windows opened data. Shaded p values represent significant effects.

	F	P	partial η^2
Automation Condition (AC)	50.50	< 0.001	0.30
Block	17.69	< 0.001	0.07
Task	518.93	< 0.001	0.69
Block * AC	7.63	< 0.001	0.06
Task * AC	81.33	< 0.001	0.41
Block * Task	17.47	< 0.001	0.07
Block * Task * AC	3.40	< 0.001	0.03

Overall Effect of Automation

The main effect of condition was evidence that the participants in the different reliability conditions opened different amounts of windows. These data are presented in Figure 4.

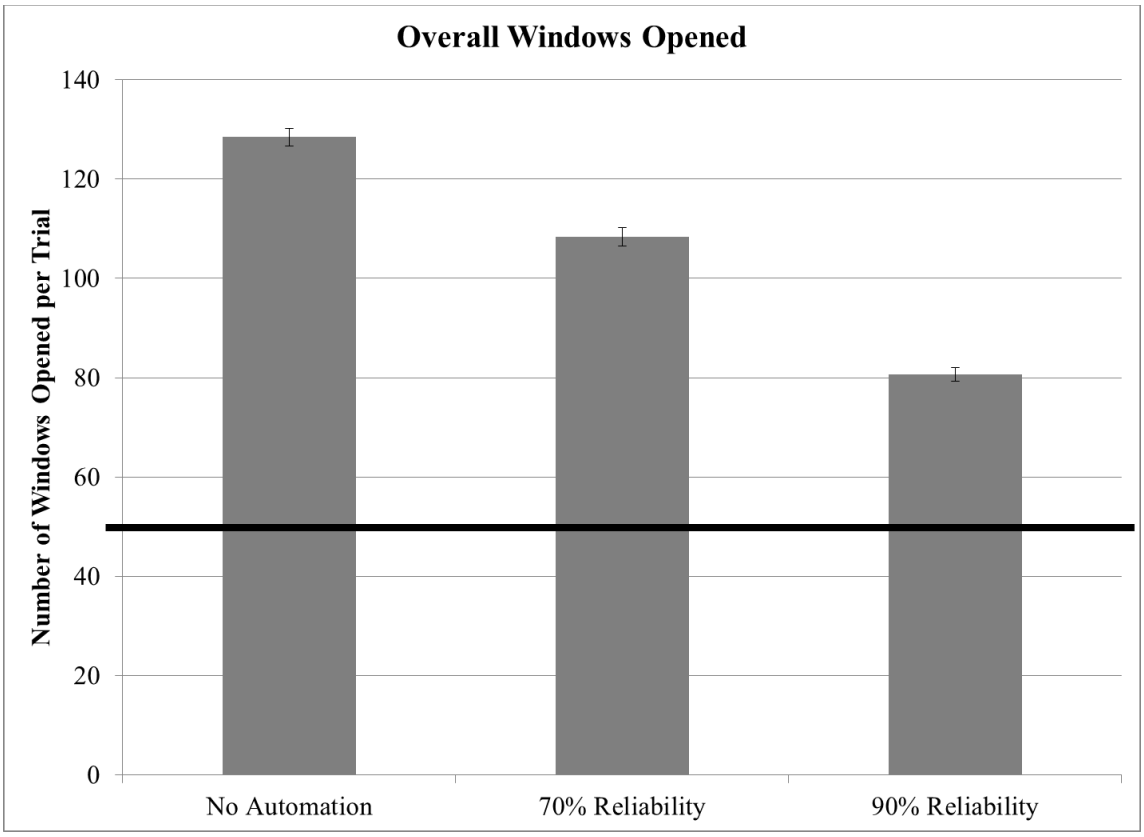

Figure 4. The mean overall number of windows opened by automation condition with standard error bars. The line across the graph at 50 represents the minimum number of windows needed to score maximum points.

Post-hoc pairwise t-tests (Bonferroni corrected) indicated significant differences between each of the three automation conditions (all p's < .001). Those in the non-automated condition opened significantly more windows than those in the automated ones, and the 70% reliable group opened significantly more windows than the 90%

group. This pattern suggests that automation reduces the level of workload by minimizing the number of windows required to be opened; the more reliable the automation, the greater the workload reduction.

Learning Effects

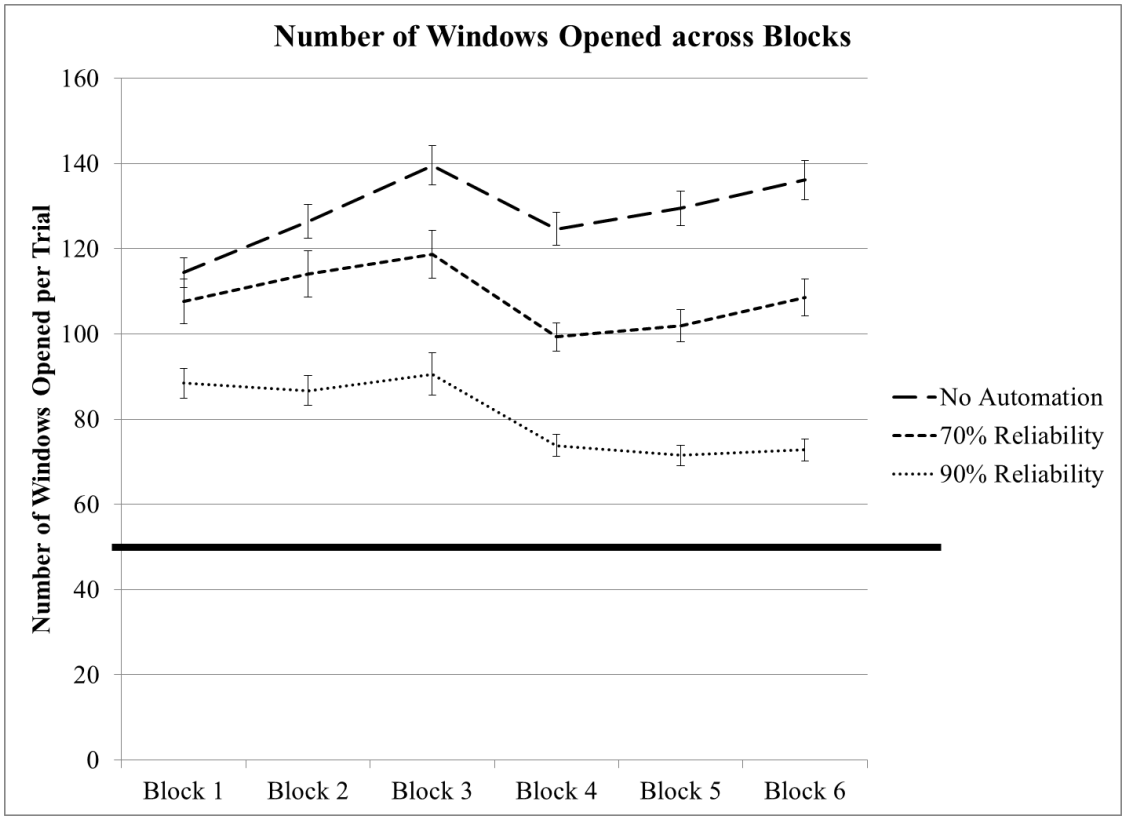

Figure 5. The number of windows opened across blocks by automation condition with standard error bars. Blocks 1-3 were on day 1. Blocks 4-6 were on day 2. The line across the graph at 50 represents the minimum number of windows needed to score maximum points.

There was an interaction of automation condition and block, as illustrated in Figure 5. The effect of automation condition persists across the blocks, but the relative

differences change early vs. late in the experiment. This pattern is clear in Figure 6 which contrasts the block 1 and block 6 data.

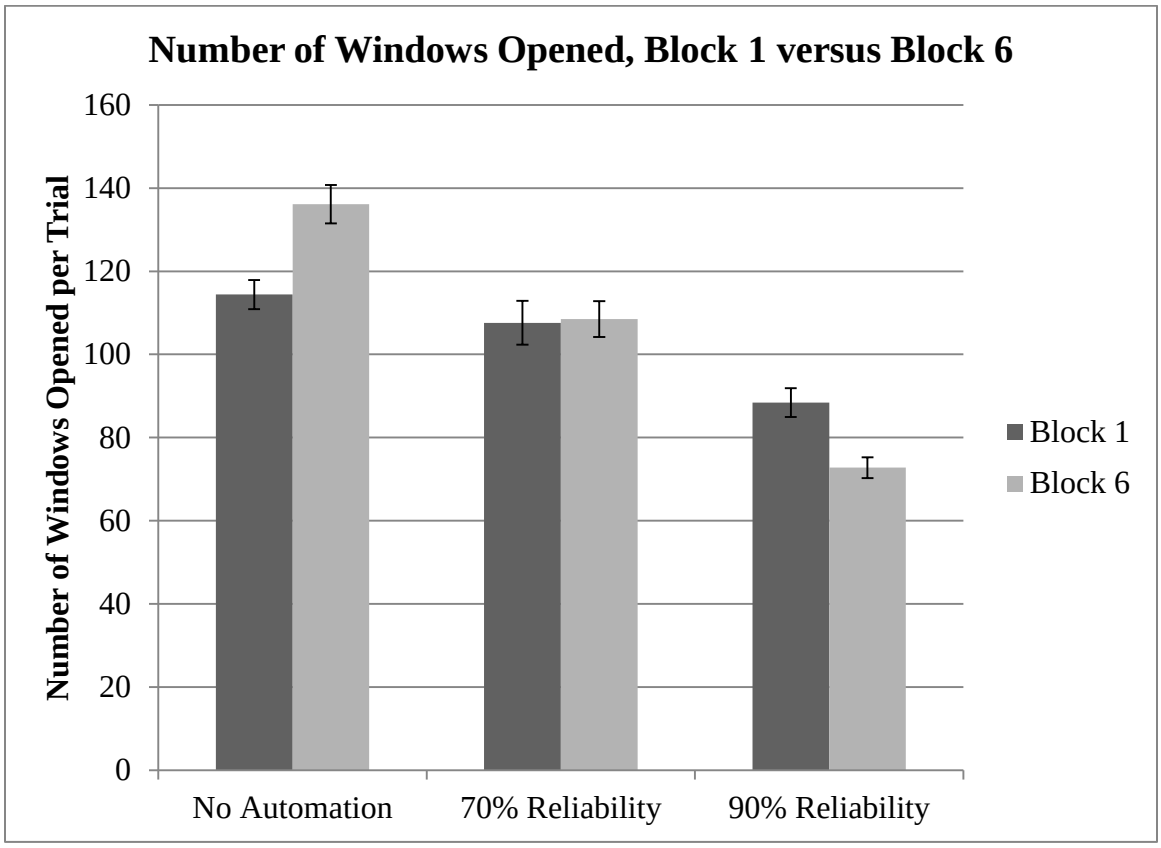

Figure 6. The number of windows opened in blocks 1 and 6 by automation condition with standard error bars.

Pairwise comparisons revealed the following patterns: participants in the no automation condition opened significantly more windows after practice ($t = -5.94$, $p < .001$), those in the 70% showed no differences ($p = .88$), and those in the 90% opened significantly fewer ($t = 4.55$, $p < .001$). Automation, then, seemed to provide a way for the participants to learn to hold steady or lessen their workload over time, while those

participants without the automated aid continued to increase workload in an effort to increase performance.

Allocation of Visual Attention across Tasks

To understand how the automation affected the way in which participants in different reliability conditions interacted with the system, I analyzed the task effect and its interactions. The main effect of task was not unexpected, as the tasks had different demands, different frequencies, and different point structures. Of more interest to the goals of this study was the interaction of task and reliability condition, plotted in Figure 7.

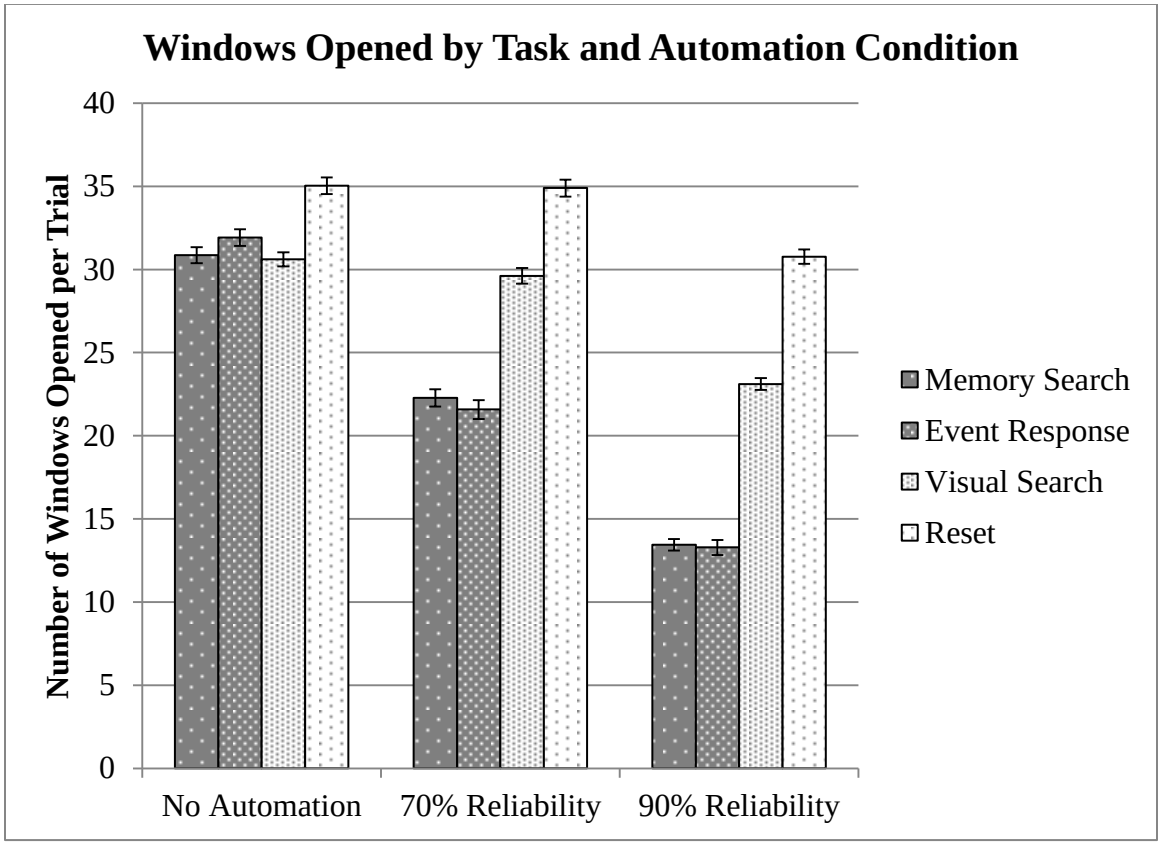

Figure 7. The number of windows opened across all blocks by task and automation condition with standard error bars. Within each condition cluster, the tasks are grouped

by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

Table 5. Pairwise comparisons on windows opened between the different tasks for each of the different automation conditions for all experimental trials. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	p	t	p	t	p
Memory Search vs. Event Response	-2.79	.006	2.56	0.011	0.51	0.608
Memory Search vs. Visual Search	0.82	0.412	-23.15	< .001	-30.54	< .001
Memory Search vs. Reset	-10.57	< .001	-36.65	< .001	-47.48	< .001
Event Response vs. Visual Search	3.69	< .001	-25.35	< .001	-33.68	< .001
Event Response vs. Reset	-8.21	< .001	-40.19	< .001	-51.63	< .001
Visual Search vs. Reset	-14.84	< .001	-18.58	< .001	-34.48	< .001

Task comparisons using paired t-tests are shown in Table 5. Participants in the no automation condition opened the window of the reset task the most, then the event response, and the two other tasks third most (with no significant difference between them. This pattern does not suggest any preference for task based on the criticality or frequency.

Participants in the 70% and 90% reliable conditions showed a different pattern than the no automation condition: the same patterns, with the reset task being opened the most, followed by the visual search task, followed by the two high-criticality/low-frequency tasks in third with no significant differences between them. This shows some matching to the frequency of the tasks, with those tasks that happen more often being checked more.

These effects, however, are compounded by the fact that they are compiled across six blocks of trials. To see how the effects differed between the beginning and end of the experiment, I tested blocks 1 and 6. Figure 8 shows these data for block 1 whereas Table 6 shows the analyses.

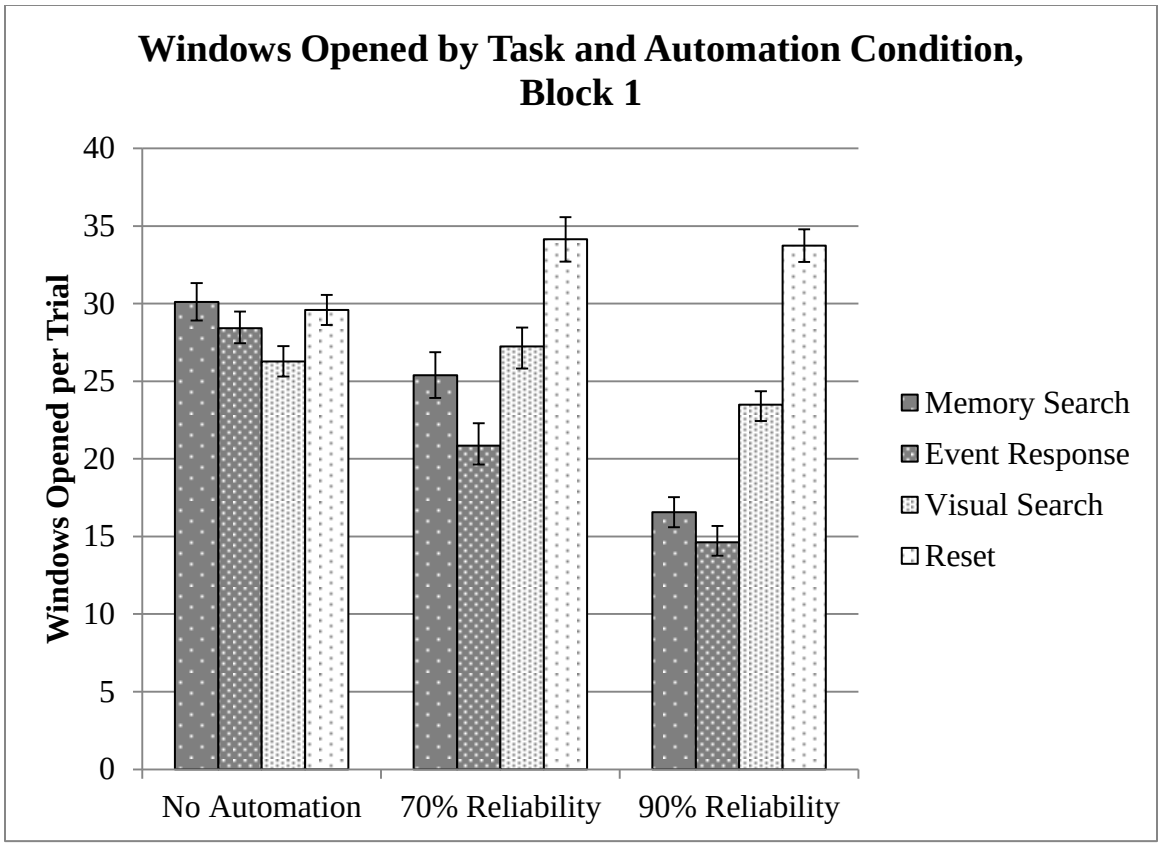

Figure 8. The number of windows opened by task and automation condition in block 1 with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

Table 6. Pairwise comparisons on windows opened between the different tasks for each of the different automation conditions in block 1. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	p	t	p	t	p
Memory Search vs. Event Response	1.42	0.160	6.72	< .001	2.39	0.019
Memory Search vs. Visual Search	4.85	< .001	-2.39	0.019	-7.63	< .001
Memory Search vs. Reset	0.62	0.533	-10.48	< .001	-17.96	< .001
Event Response vs. Visual Search	1.90	0.062	-8.06	< .001	-13.12	< .001
Event Response vs. Reset	-0.99	0.326	-17.50	< .001	-22.68	< .001
Visual Search vs. Reset	-5.59	< .001	-10.72	< .001	-20.07	< .001

I compared each task to every other task within the conditions, for a total of 6 paired t-tests per condition. Certain trends emerge early in practice. First, there were only a few differences between the allocations of those participants with no automation. They checked the memory search and reset tasks significantly more than the visual search task, but the type of task had no effect on the number of windows opened.

In the 90% reliable condition, however, the two low-criticality/high-frequency tasks are checked the most often, with the reset task being the most prominent. The two high-criticality/low-frequency tasks (memory search and event response) were checked about as often as each other, with no significant differences between them. This pattern suggested that the automation supported a more efficient checking strategy; the tasks checked the most were the ones that happened the most.

In the 70% reliable group, the reset task was checked significantly more than any other task, followed by the memory and visual search tasks together, then the event response task. Looking at the data from the other two conditions, this seemed to be a graded effect; the most-checked task is a low-criticality/high-frequency one, whereas the least-checked is the opposite. The other variance in the data was probably due to the fact that these were early trials, and the participants were still getting used to the operation of the system.

By block 6, the effects observed in block 1 were clearer. Figure 9 shows the data whereas Table 7 shows the analyses. By the end of practice participants in the 70% and 90% conditions were opening the windows of the two low-criticality/high-frequency tasks the most, with the reset task on top. Again, for both, the two high-criticality/low-frequency tasks were allotted similar numbers of windows. This means that the

participants in the 70% condition had gained a graded version of the efficient checking strategy that the 90% condition exhibited in both block 1 and block 6.

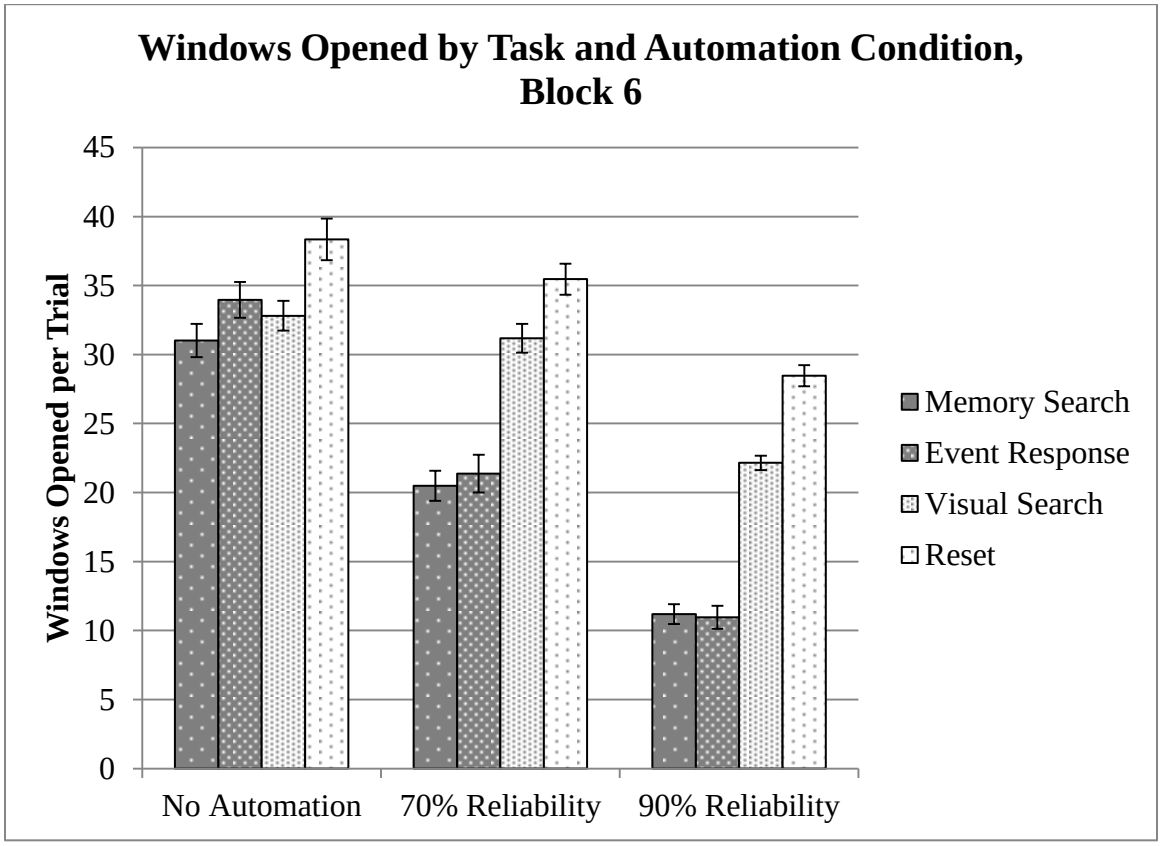

Figure 9. The number of windows opened by task and automation condition in block 6 with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

Table 7. Pairwise comparisons on windows opened between the different tasks for each of the different automation conditions in block 6. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	p	t	p	t	p
Memory Search vs. Event Response	-3.74	< .001	-1.44	0.153	0.66	0.513
Memory Search vs. Visual Search	-2.21	0.030	-14.43	< .001	-21.78	< .001
Memory Search vs. Reset	-6.59	< .001	-19.18	< .001	-27.23	< .001
Event Response vs. Visual Search	1.31	0.194	-12.66	< .001	-19.37	< .001
Event Response vs. Reset	-4.47	< .001	-18.44	< .001	-25.15	< .001
Visual Search vs. Reset	-6.48	< .001	-6.33	< .001	-13.33	< .001

In the no automation condition, the reset task was checked significantly more than all the others. The memory search task also garnered more visual attention than the event response task.

The effects not explained by the criticality/frequency manipulation, such as the fact that the reset task was higher for all conditions, may be due to the amount of information the different tasks provided. The reset task continually provided information, even when it did not need a response. Participants may have been reinforced by opening that task; always learning something when the task was opened. In comparison, the other three tasks (memory search, visual search and event response) only had one active state and did not provide any information when not active.

Summary

The automated aid alleviated participants' workload by reducing the number of windows they had to open. This effect varied over time: the more reliable the automation, the more experience alleviated workload. In all cases, the effects of automation were graded; the 70% reliable automation provided some benefit over no automation at all, and the 90% provided some benefit over the 70%.

The effects of the different task attributes on the allocation of windows opened to those tasks were striking. Allocation strategies that matched the frequency of the tasks emerged in the automated conditions, with the participants in the 90% reliable condition adopting the strategy early on in the process and those in the 70% automation following suit in the end. Task attributes other than criticality and frequency dictated some of the

allocations, however, as the continual information provided by the reset task seemed to have increased the number of times it was checked.

To understand how these allocation strategies fit into a larger view of the system and the participants' performance in it, these data must be supplemented with the task performance of the participants. An alleviation of workload and a shift in visual attention allocation can only be fully understood in the context of their effects on performance.

Points Scored

The overall number of points scored across a trial provided a measure of overall task performance; the participants' goal across all trials was to maximize their point score. As the participants were told to maximize score for each trial, more points in every instance would be preferable. Each task was able to award a maximum of 600 points per trial, so the maximum number of points available in a trial overall was 2400.

Due to the interaction of each task's score with the number of windows opened (the participants lost two points on a task every time they opened a window), the point scores were corrected by adding the points lost to window opening back into the task and overall trial scores. More information on this interaction and how it the performance score was adjusted is in Appendix J.

The results of the three-way mixed ANOVA measuring points scored, crossing automation condition (none, 70% reliable, and 90% reliable), task (memory search, visual search, reset, and event response), and block (blocks 1-6) are shown in Table 8. The analysis of the two within-participants variables violated Mauchly's sphericity test ($p < 0.001$), so the Greenhouse-Geisser correction was used in all cases where the task and

block variables were involved. All follow-up pairwise comparisons were Bonferroni corrected at the level of $\alpha = .05/$(the number of analyses done).

Table 8. Three-way mixed ANOVA for the corrected points scored data. Shaded p values represent significant effects.

	F	p	Power	partial η^2
Automation Condition (AC)	33.52	< 0.001	> .99	0.22
Block	14.26	< 0.001	> .99	0.06
Task	46.22	< 0.001	> .99	0.16
Block * AC	2.76	0.005	0.94	0.02
Task * AC	3.87	0.003	0.93	0.03
Block * Task	1.67	0.097	0.76	0.01
Block * Task * AC	1.40	0.128	0.88	0.01

Overall Effect of Automation

As with windows, the first effect of interest in the task performance measure was automation condition, to see whether there was a benefit of automation. Figure 10 shows overall points scored by each automation condition.

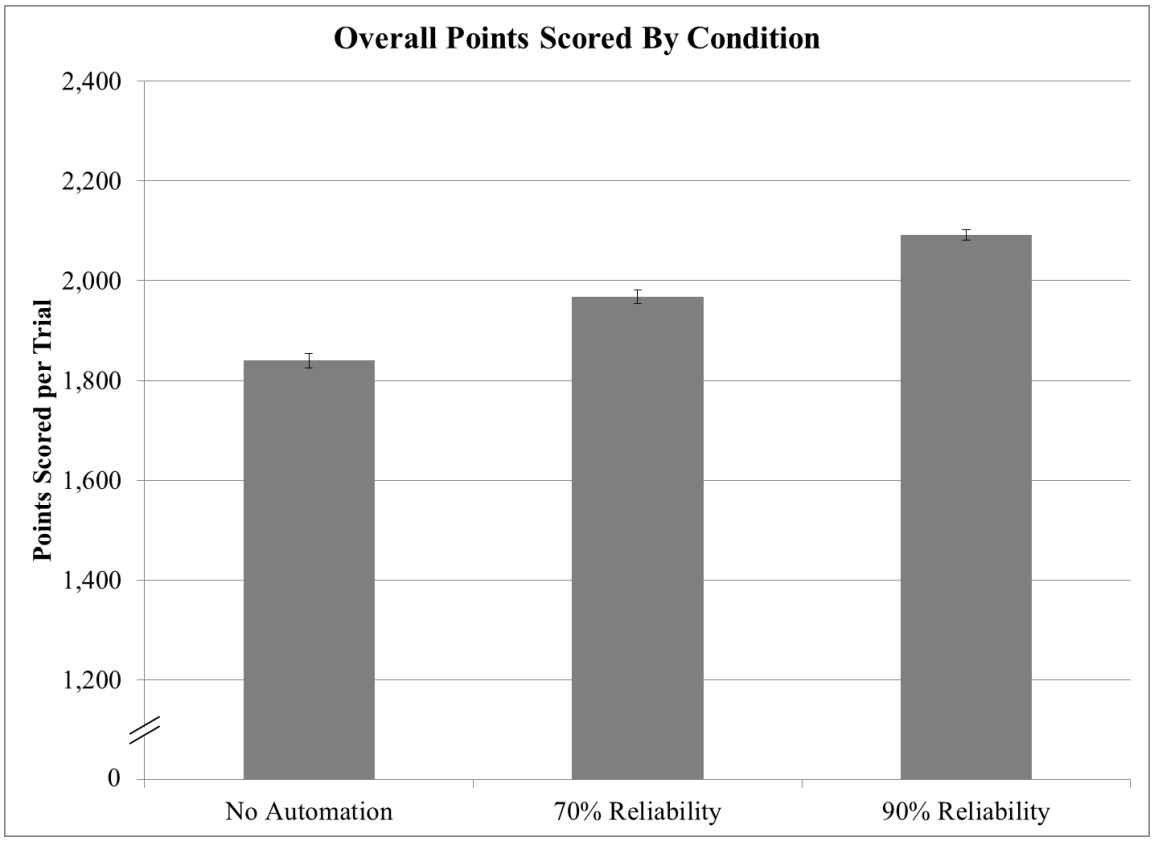

Figure 10. The overall number of corrected points scored for each of the automation conditions with standard error bars. A score of 2400 is the maximum possible number of points in any given trial.

To better understand the main effect of automation condition, I conducted pairwise comparisons between the different conditions. All three were significantly different from each other ($p < .001$), with those in the 90% scoring the most, those in the 70% in the middle, and the non-automated condition participants scoring the least. This means that the benefit of automation was not only significant, but graded. 70% automation helped over none, and 90% helped more than 70%.

Learning Effects

To understand better how the participants fared as they became more experienced with the system, I analyzed the differences among the blocks. Figure 11 shows the

numbers of points scored across blocks for each condition, whereas Figure 12 shows the performance at the beginning (block 1) and end (block 6).

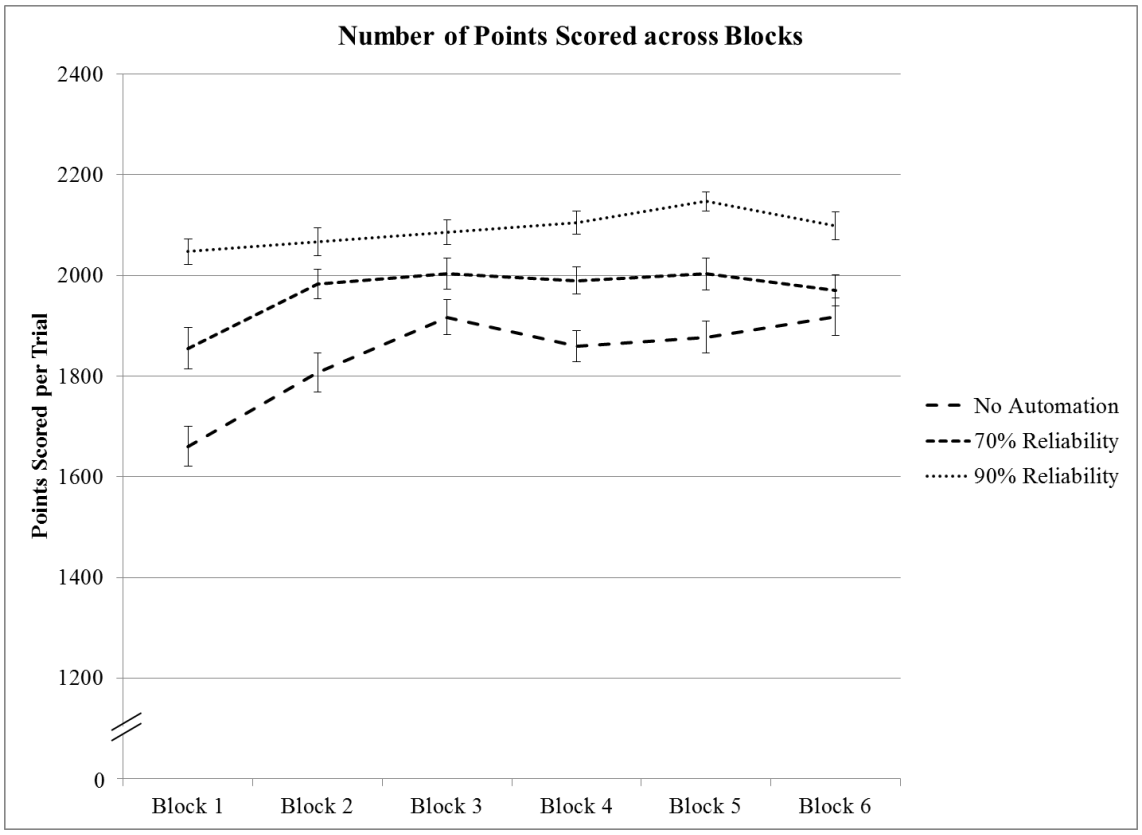

Figure 11. The number of points scored across blocks by automation condition with standard error bars. Blocks 1-3 were on day 1. Blocks 4-6 were on day 2. 2400 is the maximum score possible in any specific trial.

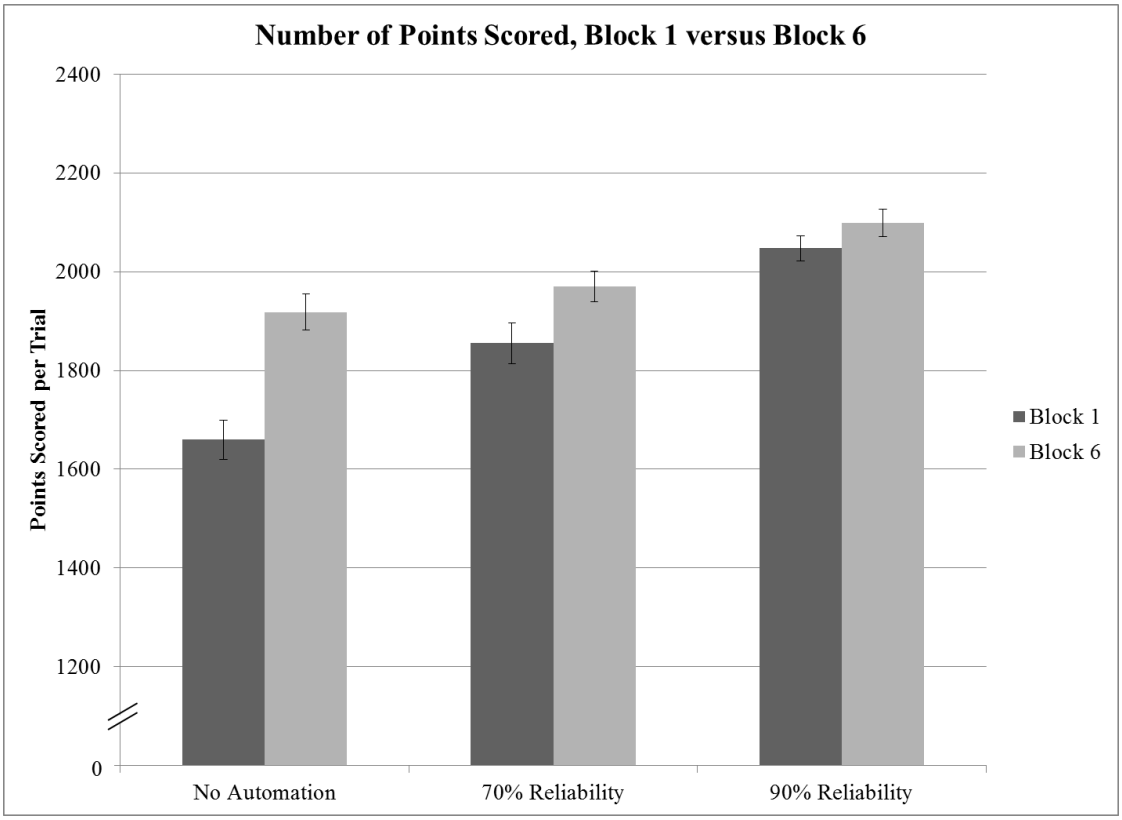

Figure 12. The number of points scored in blocks 1 and 6 by automation condition with standard error bars. A score of 2400 is the maximum score possible in any specific trial.

To understand how the effect of experience affected the conditions differently, I ran paired t-tests for each condition comparing the block 1 levels with the block 6 ones. Participants from both the no automation and the 70% reliable conditions significantly improved from block 1 to block 6 ($t = -5.75$, $p < .001$ and $t = 3.18$, $p = .001$, respectively), whereas the 90% condition showed no significant improvement ($t = -1.50$, $p = .137$).

To see how the conditions compared to each other, I also performed independent pairwise comparisons of the three conditions at block 6. This was done due to the assertion that the 70% condition was chosen because it would provide neither a benefit

nor detriment to performance (Wickens & Dixon, 2007). The 90% reliable condition performed significantly better than either other condition ($t = -3.922$, $p < .001$ for no automation; $t = -3.081$, $p = .002$ for 70% reliable), but the 70% reliable and non-automated conditions performed at a similar level ($t = -1.082$, $p = .281$). This means that, at the end of the experimental trials, 70% reliable automation did not show a significant benefit of performance over no automation.

Allocation of Points across Tasks

The ANOVA showed a main effect of task, suggesting that participants performed differently on different tasks. Again, this is expected, as the different tasks were not matched in difficulty and demands. Of more interest was the interaction of task and automation condition, shown in Figure 13.

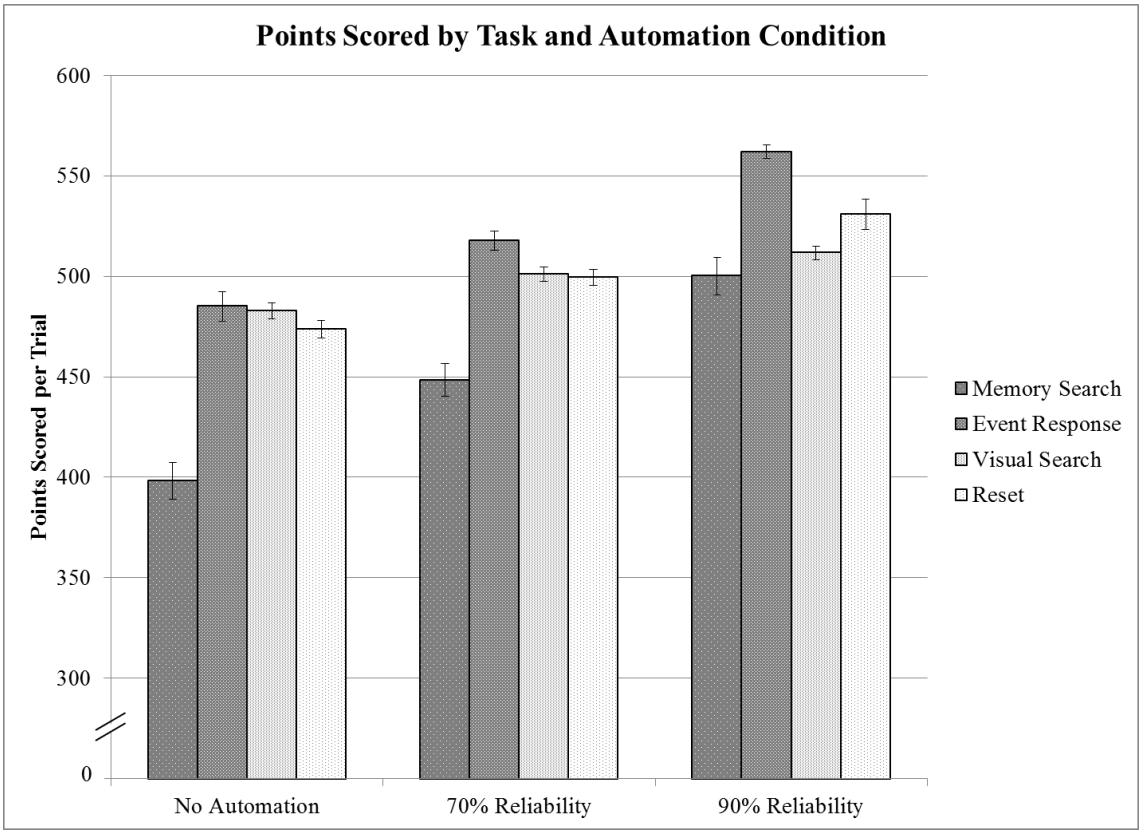

Figure 13. The number of points scored across all blocks by task and automation condition with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right. A score of 600 is the maximum number of points possible on any one task in any one trial.

Table 9. Pairwise comparisons on points scored between tasks for the different automation conditions. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	p	t	p	t	p
Memory Search vs. Event Response	-8.17	< .001	-8.28	< .001	-6.34	< .001
Memory Search vs. Visual Search	-9.05	< .001	-6.74	< .001	-1.25	0.212
Memory Search vs. Reset	-7.64	< .001	-5.53	< .001	-2.62	0.009
Event Response vs. Visual Search	0.30	0.766	3.19	0.002	10.30	< .001
Event Response vs. Reset	1.38	0.169	2.93	0.004	3.78	< .001
Visual Search vs. Reset	1.73	0.085	0.35	0.729	-2.43	0.015

I conducted pairwise analyses between each set of tasks to determine where the differences laid (see Table 9). The memory search task was consistently lower for all three automation conditions. In the non-automated condition, participants performed similarly in three tasks, but significantly more poorly on the memory search task. Those participants in the 70% reliable condition performed best on the event response task, worst on the memory search task, and similarly on the other two. In the 90% reliable condition, participants performed significantly better on the event response task, and similarly on the other three. These effects show that, the more reliable the automation, the relatively better the participants performed on the high-criticality/low-frequency tasks.

Because the task x block and task x block x automation condition interactions were not significant, there were no reasons to delve into the differences between tasks in blocks 1 and 6, as they would show similar effects.

Summary

The performance data indicated that reliable automation was beneficial. Those participants in the 90% reliable condition consistently outperformed the other two conditions. Those in the 70% reliable condition started off performing better than those with no automation, but the block 6 data showed no significant differences in performance at the end.

The differences between tasks showed that the high-criticality/low-frequency tasks benefitted relatively more the addition of automation, boosting relative levels of performance for both compared to the other tasks.

Efficiency

The two non-derived measures (windows opened and points scored) each captured a specific part of the data. The number of windows opened provided a measure of workload. The number of points scored provided a measure of task performance. I created an efficiency score to measure how the benefits of workload alleviation informed the difference in points; how less work and better performance come together to create a more complete view of the effects of automation in multiple-task systems.

Efficiency was calculated using the following formula:

Equation 1. The equation used to calculate efficiency score for each task and for the trial overall in this study.

$$Efficiency = \frac{Number\ of\ Corrected\ Points\ Scored}{Number\ of\ Windows\ Opened}$$

It was calculated for every task and the trial overall. Because points were in the numerator and windows were in the denominator, higher efficiency wes better. An efficiency score of 48 was the maximum possible for a single trial (2400 possible points/50 windows opened minimum to attain that score).

The results of the three-way mixed ANOVA measuring points efficiency, crossing automation condition (none, 70% reliable, and 90% reliable), task (memory search, visual search, reset, and event response), and block (blocks 1-6) are shown below in Table 9. Power for all effects was above .99. The analysis of the two within-participants variables violated Mauchly's sphericity test ($p < 0.001$), so the Greenhouse-Geisser correction was used in all cases where the task and block variables were involved.

Table 10. Three-way mixed ANOVA for the efficiency data. Shaded p values represent significant effects.

	F	P	partial η^2
Automation Condition (AC)	152.61	< 0.001	0.90
Block	28.41	< 0.001	0.11
Task	224.96	< 0.001	0.49
Block * AC	12.88	< 0.001	0.10
Task * AC	91.26	< 0.001	0.44
Block * Task	9.36	< 0.001	0.04
Block * Task * AC	4.57	< 0.001	0.04

Overall Effect of Automation

A high efficiency score means the participant was able to perform well while minimizing workload. Because automation has been shown to both alleviate workload and improve performance, the effect on efficiency should have been significant.

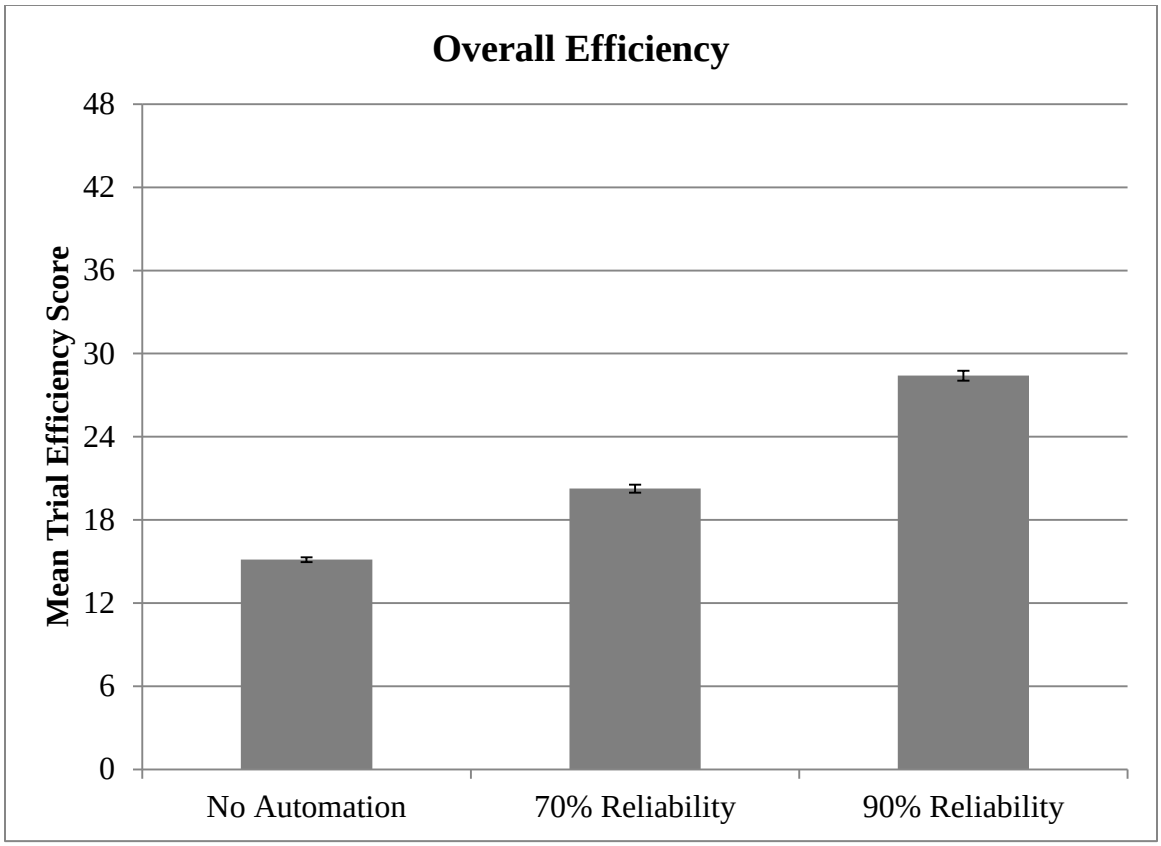

Figure 14. The overall efficiency by automation condition with standard error bars. An efficiency score of 48 is the maximum possible overall efficiency for a trial.

The efficiency score data by automation condition are shown in Figure 14. I first analyzed the effect of automation condition on efficiency, conducting pairwise independent t-tests on the different conditions. All differences were significant ($p < .001$), meaning that those in the 90% reliable condition were the most efficient, followed by those in the 70%, and then those with no automation, in that order. This follows the results found in the other two measures, as they opened fewer windows and scored more points.

Learning Effects

Given that the block-by-block learning effects for windows and score show different results (such as the non-automated conditions opening more windows *and* scoring more points across the blocks whereas the 90% reliable condition opened fewer and had no point change), I ran the same analysis for efficiency. I felt that this analysis would be important to best understand how the different conditions differed with experience. The efficiency scores for each condition across blocks are shown in Figure 15. As with windows and points, I decided to compare block 1 directly against block 6 to see how the different automation conditions fared at the beginning and end of the experimental trials. Figure 16 shows the data for these two blocks.

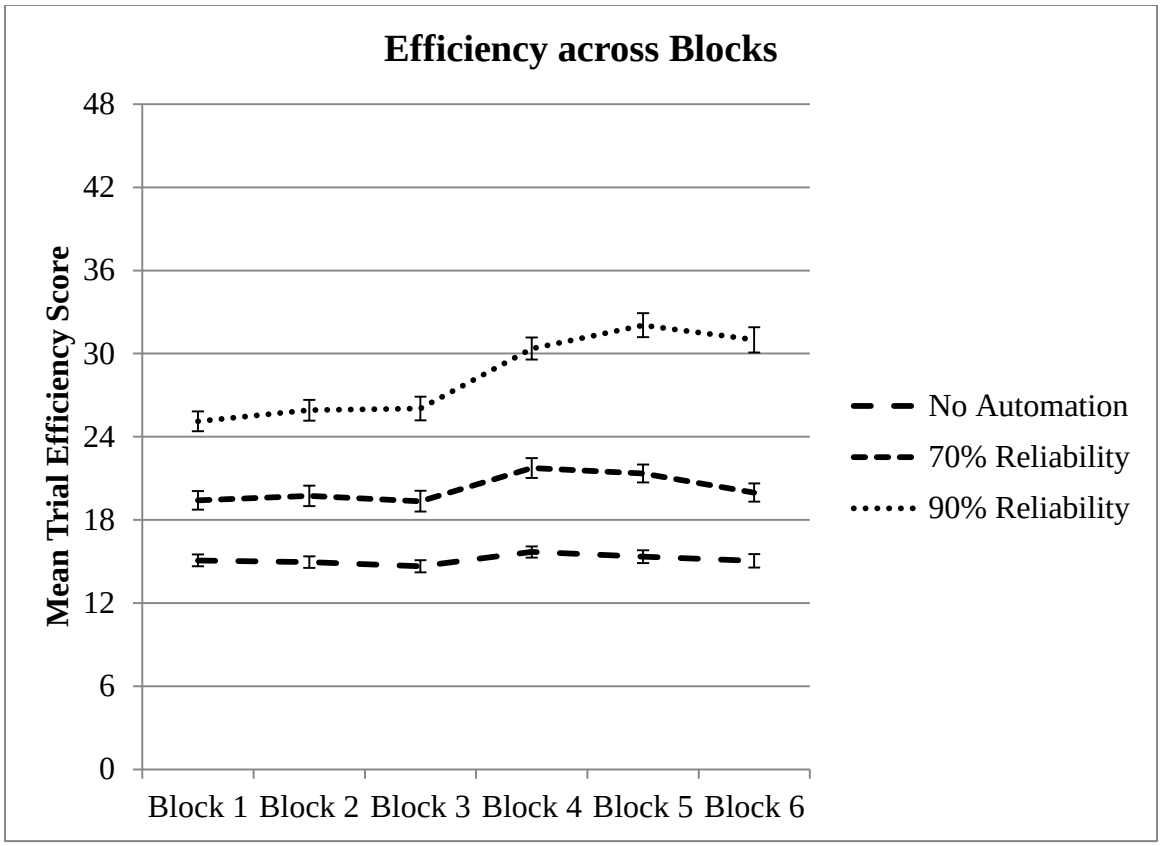

Figure 15. Efficiency across blocks by automation condition with standard error bars. Blocks 1-3 were on day 1. Blocks 4-6 were on day 2. An efficiency score of 48 is the maximum efficiency score possible in any specific trial.

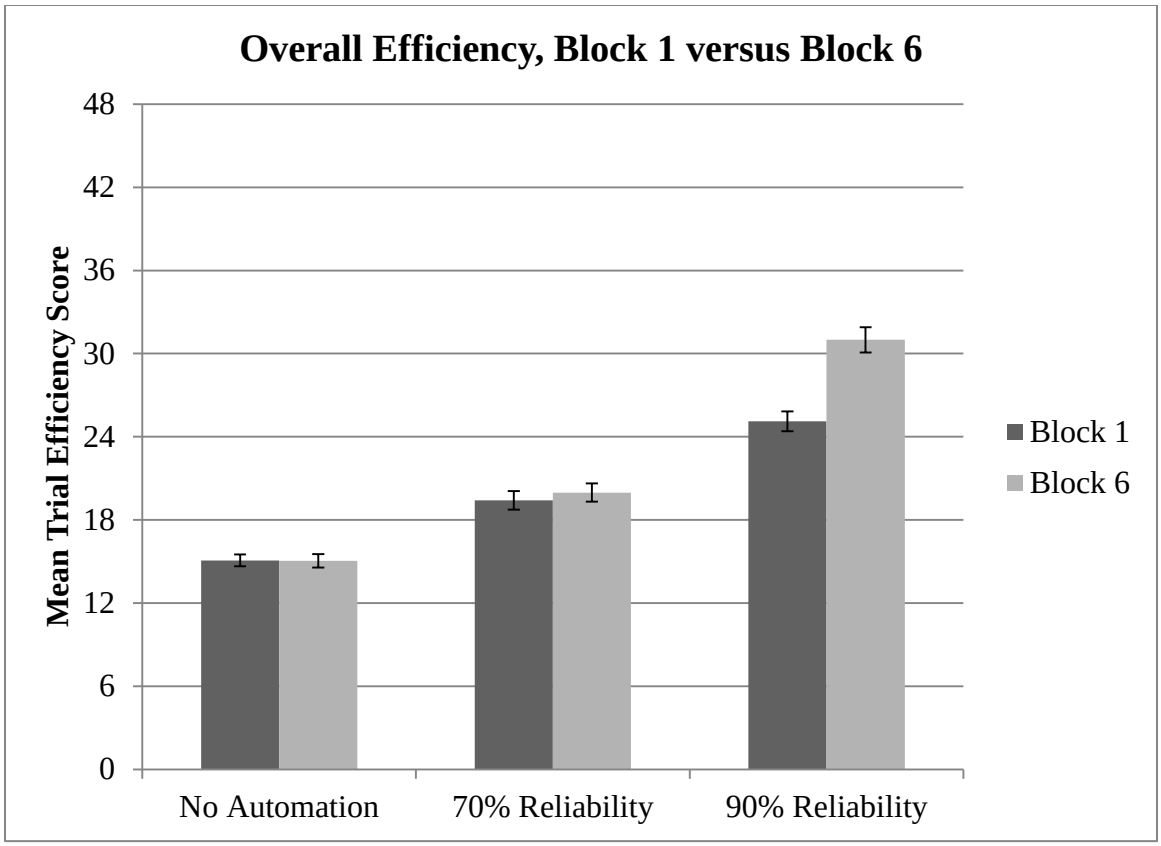

Figure 16. Efficiency in blocks 1 and 6 by automation condition with standard error bars. An efficiency score of 48 is the maximum efficiency score possible in any specific trial.

To test the interaction of block x automation condition, I ran paired t-tests for each condition comparing efficiency in block 1 to efficiency in block 6. Whereas the 90% reliable condition showed significant improvement ($t = -5.63$, $p < .001$), both the 70% reliable condition and the non-automated condition showed no significant improvement ($t = -.72$, $p = .476$; $t = .22$, $p = .826$, respectively). This means that not only does highly reliable automation start at a higher level of efficiency; it supports gains in efficiency with practice. Lower levels of reliability do not offer this practice gain.

Efficiency across Tasks

The analysis of the main effect of task in the efficiency ANOVA comes with two pitfalls: One, the maximum efficiency for an individual task is not the same as the optimum efficiency for that task when the overall trial-wide efficiency is at a maximum. For example, if a participant opened each task window once and then spent the entire trial concentrating on the memory search task, his or her efficiency scores for the four tasks in that trial would have been: memory search = 600 (600/1), event response, visual search, and reset = -300 (-300/1). The overall trial efficiency would have been -75, or $\frac{600-300-300-300}{1+1+1+1} = -\frac{300}{4}$. As the purpose of the trials was to maximize the overall score, the optimum efficiency scores for the different tasks were regarded as their efficiency when the maximum points for the trial *overall* were achieved.

Two, because of the frequency of the different tasks, the optimum efficiency varies by task. For the two high-criticality/low-frequency tasks, (5 times a trial, 120 points per time) this optimum efficiency score would have been 120 ($\frac{600\ Possible\ Points}{5\ Minimum\ Windows}$). For the two low-criticality/high-frequency tasks, the optimum efficiency score would have been 30 ($\frac{600\ Possible\ Points}{20\ Minimum\ Windows}$).

Because of these pitfalls, it makes more sense to look at the interaction of task with condition, as the important differences are between conditions, not between tasks. As such, these graphs will be arrayed differently than the points and windows graphs, with the bars being clustered by task, not by automation condition. This interaction is plotted in Figure 16.

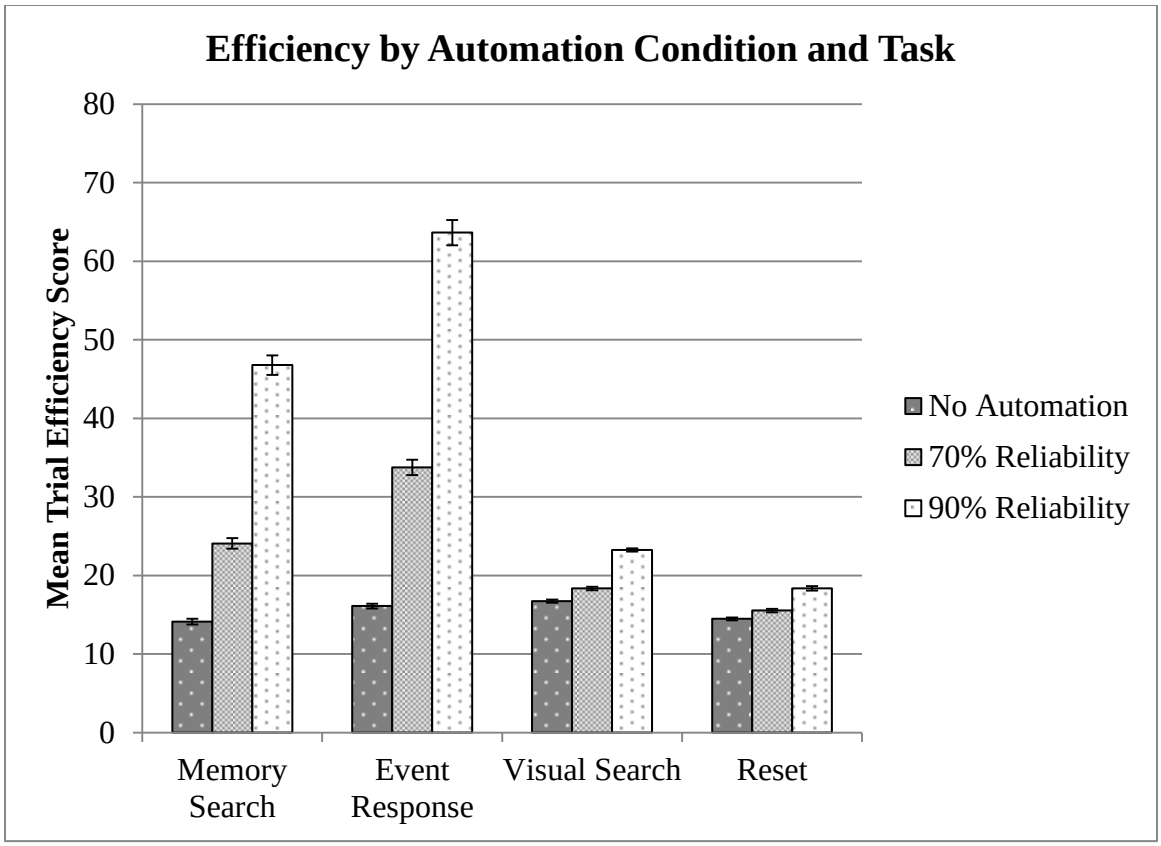

Figure 17. The number of points scored across all blocks by automation condition and task with standard error bars. For the first two task clusters, an efficiency score of 120 is the optimum. For the second two task cluster, and efficiency score of 30 is the optimum.

Table 11. Pairwise comparisons on Efficiency between automation conditions for the different tasks for all blocks combined. Shaded p values represent significant effects.

	Memory Search		Event Response		Visual Search		Reset	
	t	p	t	p	t	p	t	p
None vs 70%	-12.90	< .001	-17.22	< .001	-5.21	< .001	-3.55	< .001
None vs 90%	-25.38	< .001	-29.05	< .001	-21.50	< .001	-11.02	< .001
70% vs 90%	-16.20	< .001	-15.89	< .001	-15.47	< .001	-7.57	< .001

For all tasks, all of the comparisons were significant, with those in the 90% condition being the most efficient, those in the 70% in the middle, and those with no

automation being the least. The larger mean differences come from the two high-criticality/low-frequency tasks.

Because of the task x block interaction and the three-way interaction (task x block x automation condition), though, this may not be the whole story. I decided, then, to analyze the same data at blocks 1 and 6 to see if the effects differed from the beginning to the end. Figure 18 shows the data for the beginning (block 1) of the experimental trials whereas Table 12 shows the pairwise analysis for the same.

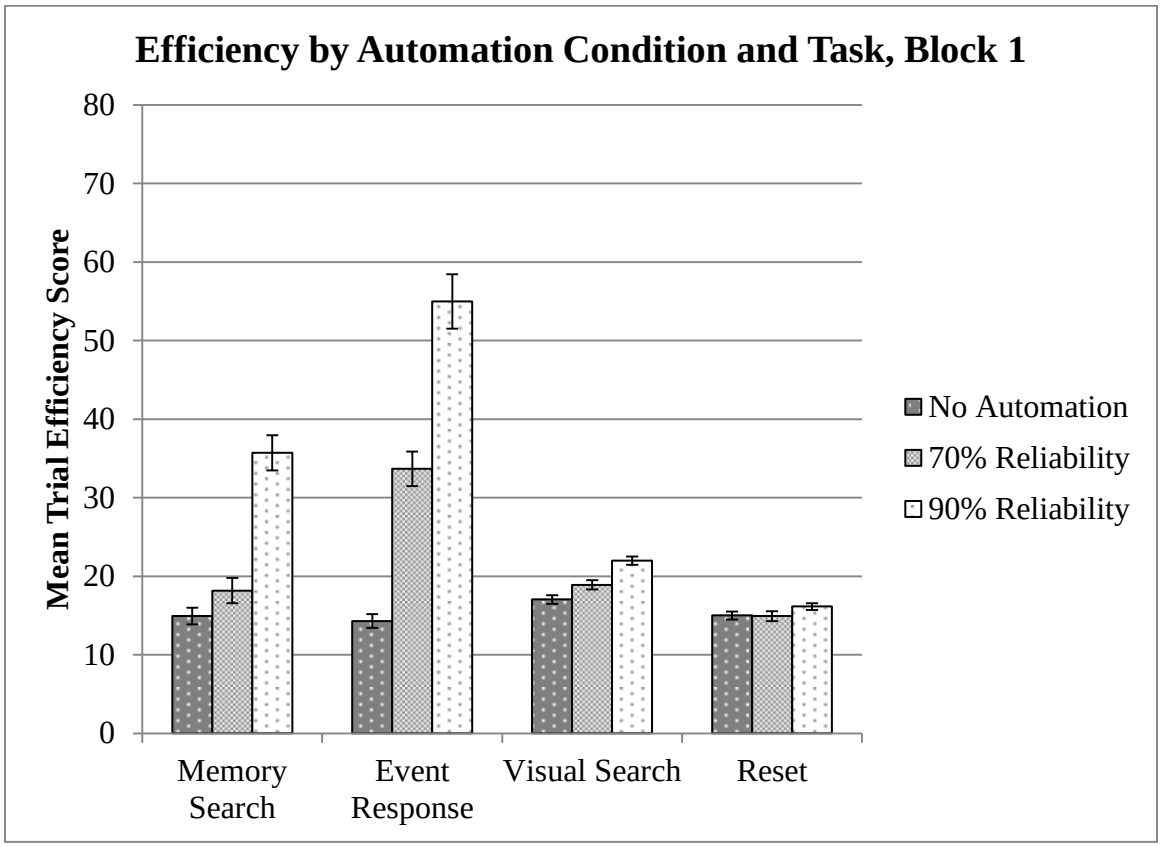

Figure 18. The number of points scored in block 1 by automation condition and task with standard error bars. For the first two task clusters, an efficiency score of 120 is the optimum. For the second two task cluster, and efficiency score of 30 is the optimum.

Table 12. Pairwise comparisons on Efficiency between automation conditions for the different tasks in block 1. Shaded p values represent significant effects.

	Memory Search		Event Response		Visual Search		Reset	
	t	p	t	p	t	p	t	p
None vs 70%	-1.68	0.094	-8.15	< .001	-2.31	0.022	0.09	0.928
None vs 90%	-8.42	< .001	-11.38	< .001	-6.53	< .001	-1.73	0.085
70% vs 90%	-6.40	< .001	-5.19	< .001	-3.90	< .001	-1.62	0.108

In block 1, at the beginning of the experimental trials, several patterns emerged. One, none of the three automation conditions showed any differences in the reset task. Second, with the exception of the event response task, the 70% reliability participants performed at the same efficiency level as their no automation counterparts. Finally, the participants in the 90% reliability condition were significantly more efficient than everyone else in everything but the reset task.

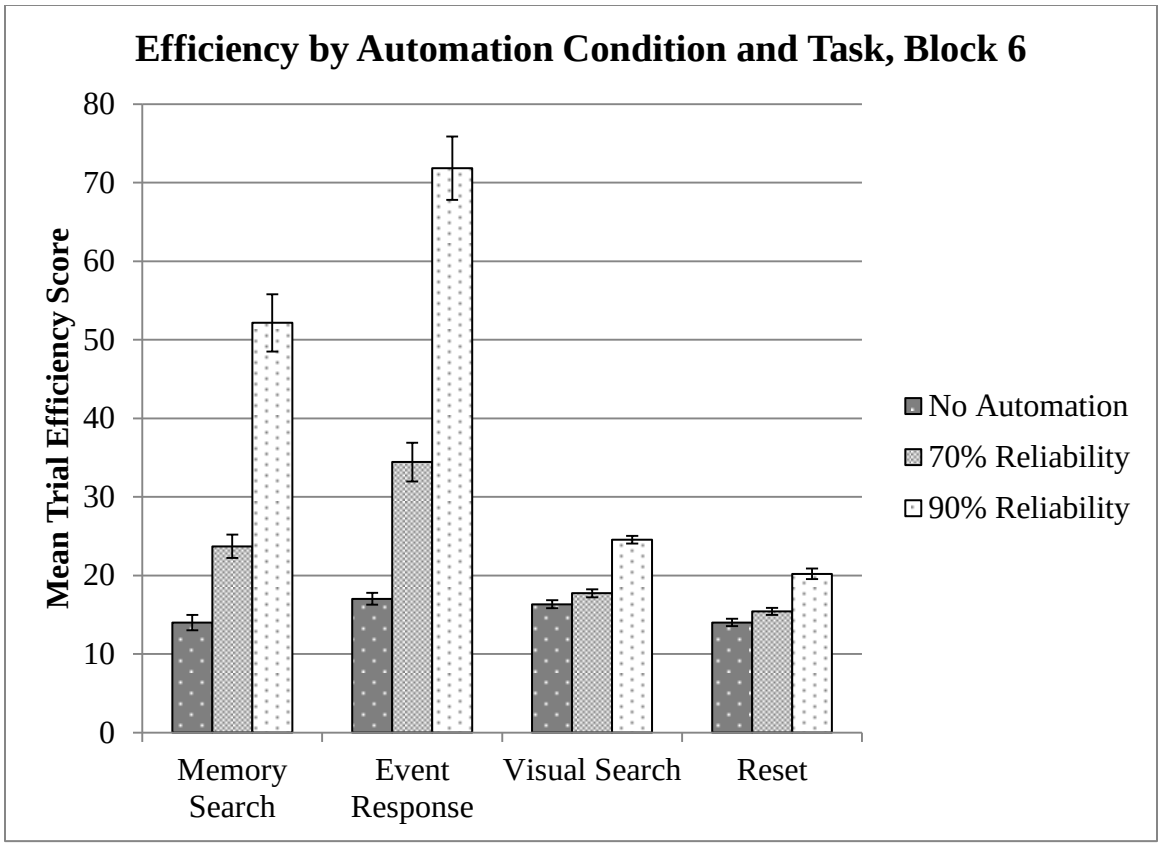

Figure 19. The number of points scored in block 6 by automation condition and task with standard error bars. For the first two task clusters, an efficiency score of 120 is the optimum. For the second two task cluster, and efficiency score of 30 is the optimum.

Table 13. Pairwise comparisons on Efficiency between automation conditions for the different tasks in block 6. Shaded p values represent significant effects.

	Memory Search		Event Response		Visual Search		Reset	
	t	p	t	p	t	p	t	p
None vs 70%	-5.49	< .001	-6.74	< .001	-1.97	0.051	-2.12	0.036
None vs 90%	-10.10	< .001	-13.40	< .001	-11.80	< .001	-7.53	< .001
70% vs 90%	-7.22	< .001	-7.93	< .001	-9.86	< .001	-5.84	< .001

The data for block 6 are shown in Figure 19, whereas the results of the analysis are shown in Table 13. The effects in block 1 became more pronounced in block 6. By

block 6, the 90% reliable condition was significantly more efficient than the other two conditions across the board. The 70% reliable condition was significantly more efficient than non-automated for the high-criticality/low-frequency tasks, but not for the low-criticality/high-frequency ones. Highly reliable automation aided efficiency in general, while lower-reliability may have aided certain task types more than others.

Summary

The compound effects of fewer windows being opened and more points being obtained provided for a more efficient use of the system for the automated conditions. These gains in efficiency were not significant for the lower level of reliability, however, as only the 90% condition improved with experience with the system.

The greatest gains in efficiency were seen for the two high-criticality/low-frequency tasks (memory search and event response). This was in part because of the higher optimal efficiency level (120 optimum efficiency versus 30 for the other two tasks), but the differences across automation conditions in these two tasks, both at the beginning and end of the experimental trials, show that the gains in efficiency that made the difference in the overall scores were these.

Transfer

At the end of the six blocks of experimental trials, all participants were transferred to a system with no automated aid for one block of four trials. The transfer trials were designed to assess how participants who had gained experience with the system using automation would react when that automated aid was taken away. Based on the results from the experimental trials, the automation provided support to both overall workload

and task performance, so there were differences between conditions that could have shown disruptions in the transfer trials.

It is important to note that, because the only change between the experimental trials and the transfer trials was the removal of the automated aid, the no automation condition experienced no change. Participants in the 70% and 90% reliable conditions were warned that the automation had been removed. The no automation participants were only told was that they were being asked to complete one more block of trials.

Because the transfer comprised four trials, I decided to do all comparisons between experimental and transfer trials using block 6 as the experimental analog. This provided for two things: One, it allowed the transfer to be compared to the most recent activity the participants had with the system; seeing how they were affected right after they had had their most experienced four trials. Second, it created a situation were similar amounts of data were being compared, as both block 6 and the transfer block were made up of four 5-minute trials.

Overall Effect of Automation

For each of the three dependent measures, I conducted paired t-tests comparing block 6 and transfer for each of the different automation conditions, using a level of $\alpha = .05$. (The efficiency analysis was not included, as it showed very similar effects without providing any more information.)

To better show the differences between block 6 and transfer, I created graphs for all measures that provided a proportion difference from the block 6 level to the transfer level. The formula for this difference score for the two measures is below.

Equation 2. The equation for calculating the proportion difference scores for windows and points for teach task and for the trial overall.

$$Proportion\ Difference = \frac{Transfer - Block\ 6}{Block\ 6}$$

Windows Opened

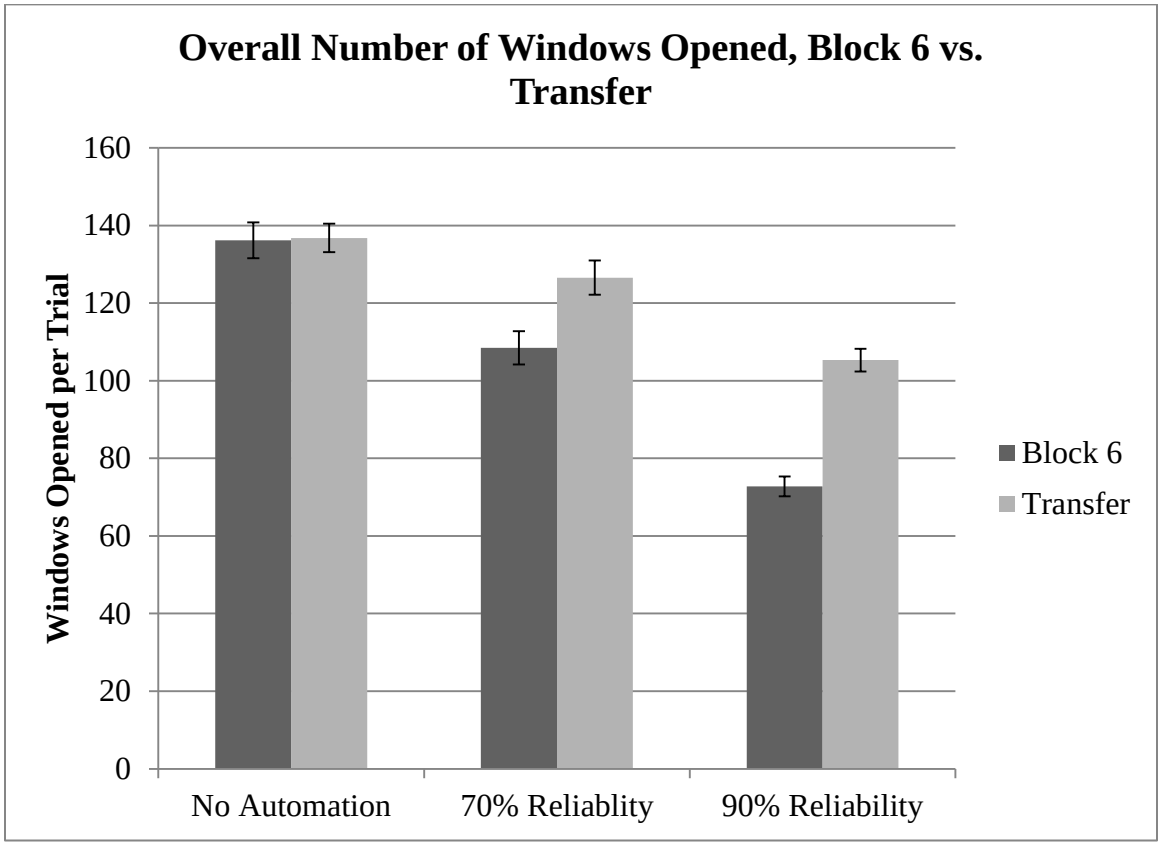

Figure 20. The overall number of windows opened by automation condition in block 6 and the transfer block with standard error bars.

Table 14. Pairwise comparisons on windows opened for each automation condition between block 6 and transfer. Shaded p values represent significant effects.

	t	p
No Automation	-0.16	0.874
70% Reliability	-7.47	< .001
90% Reliability	-12.07	< .001

The number of windows opened in block 6 versus transfer is shown in Figure 20. The results of paired t-tests comparing the three conditions' levels at block 6 and transfer is shown in Table 14. As can be seen from the results, those in the non-automated condition performed similarly before and after transfer. This was expected, as nothing changed for them. Participants in both the 70% and 90% reliable conditions opened significantly more windows, showing an increase in workload when the automation was taken away. The proportion difference is shown in Figure 21.

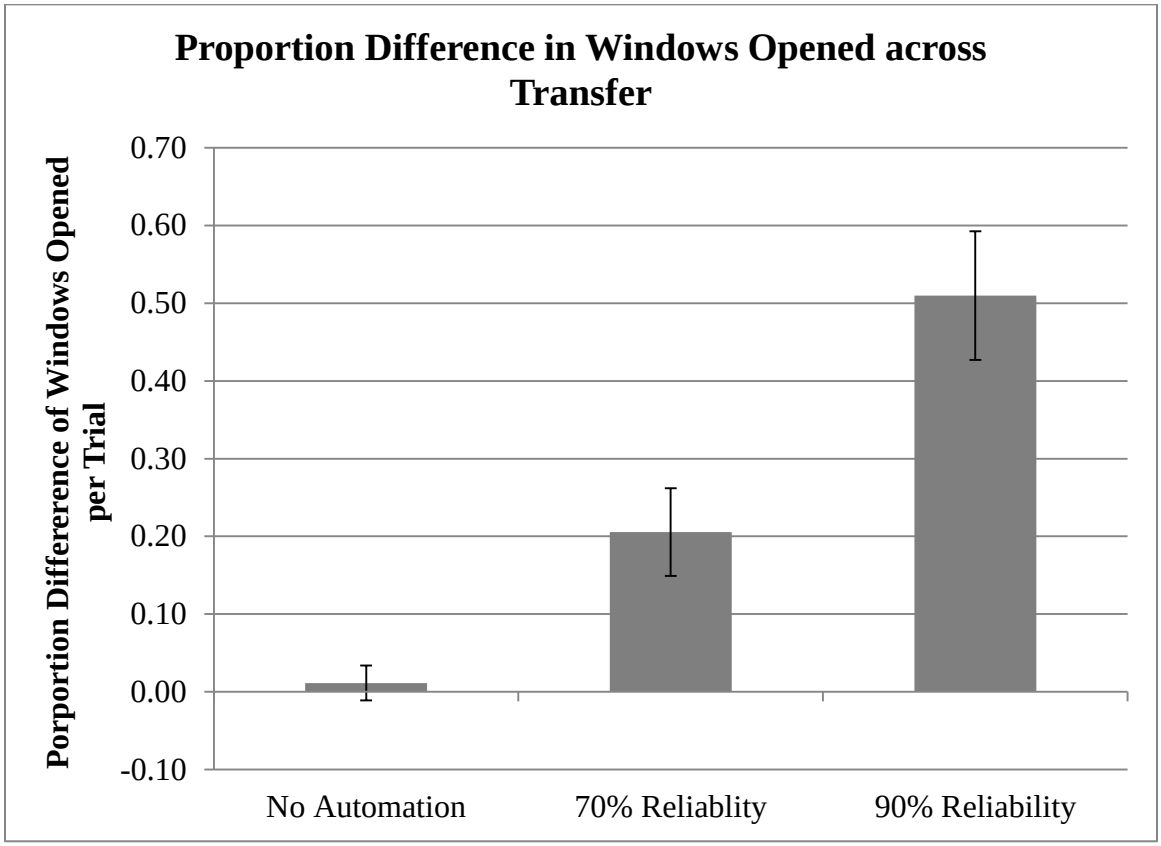

Figure 21. The proportion difference of windows opened by automation condition between block 6 and the transfer block with standard error bars.

From this graph, we can see that those in the 70% reliable condition opened approximately 20% more windows in transfer, whereas those in the 90% reliable condition opened over 50% more windows. The workload increase when the automation was taken away was significant.

Points Scored

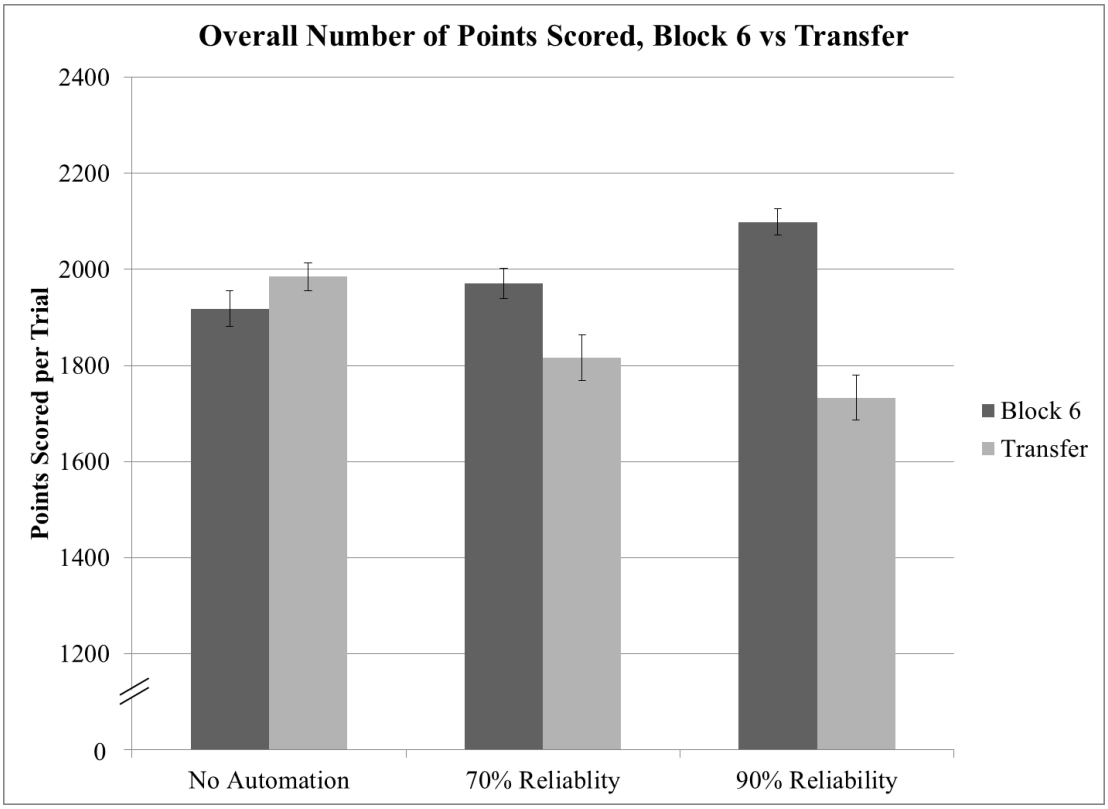

Figure 22. The overall number of points scored by automation condition in block 6 and the transfer block with standard error bars. The maximum possible score for a trial is 2400.

Table 15. Pairwise comparisons on points scored for each automation condition between block 6 and transfer. Shaded p values represent significant effects.

	t	p
No Automation	-1.88	0.064
70% Reliability	3.98	< .001
90% Reliability	7.90	< .001

The number of points scored in block 6 versus transfer is shown in Figure 22. The results of the paired t-tests are shown below in Table 15. As can be seen from the results, those in the non-automated condition performed similarly before and after transfer. This was expected. Those in both the 70% and 90% reliable conditions scored significantly fewer points, showing a decrease in performance when the automation was taken away. The proportion difference is shown in Figure 21.

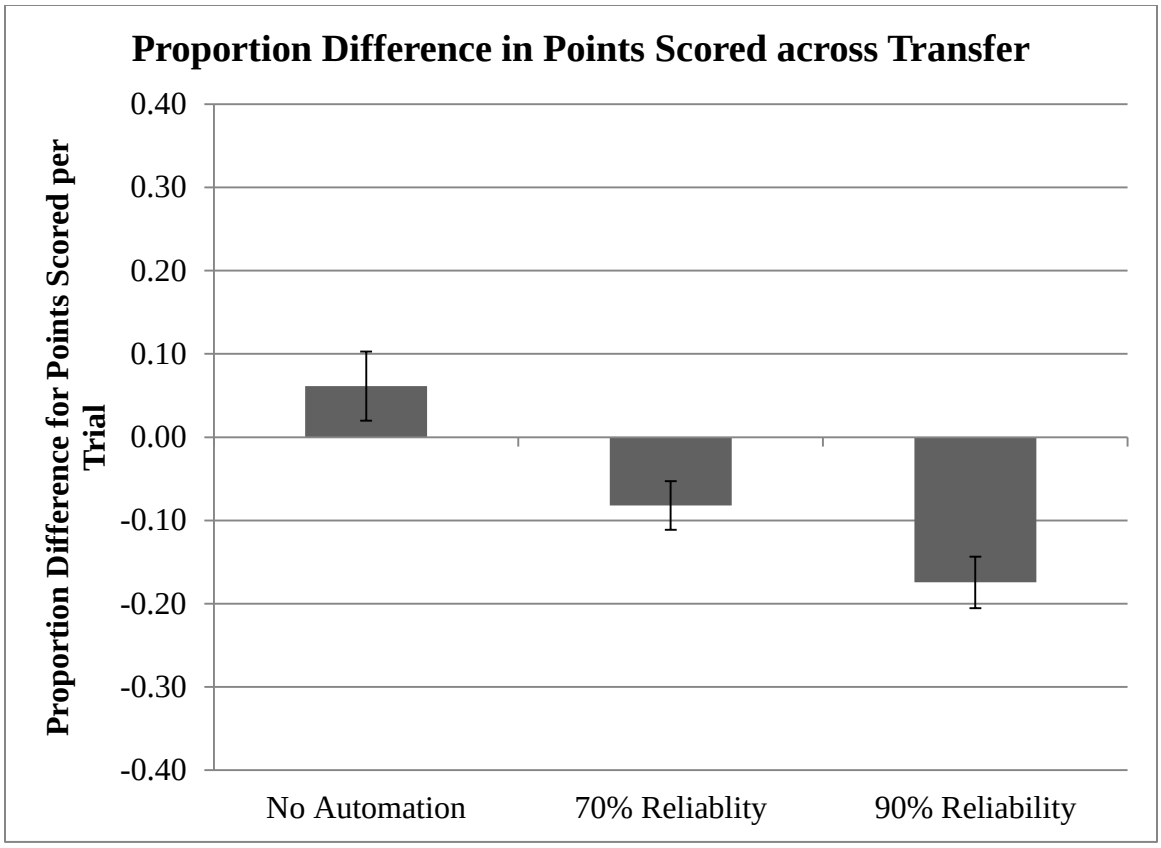

Figure 23. The proportion difference of points scored by automation condition between block 6 and the transfer block with standard error bars.

The difference in the non-automated condition is not significant. The 70% reliable participants are scoring 5-10% less, while the 90% reliable participants are scoring 15-20% less.

Summary

When the participants were transferred to a system with no automation, participants in the 70% and 90% reliable conditions suffered, both due to increased workload and worsening task performance. As they did not have automation to begin with, those in the non-automated condition stayed the same.

Differential Task Effects

In the experimental trials, differences between tasks provided information about how automation at varying levels of reliability affected the way the participants allocated their visual attention between separate tasks and how that, in turn, affected their task performance. These next analyses were to assess whether removing the automated aid changed those effects. For the windows and points measures, I compared the levels at block 6 and transfer, using the Bonferroni-corrected level of $\alpha = .05/4 = .013$. For each measure, I also created a proportion difference plot to show which tasks were most affected by the loss of automation.

Windows Opened

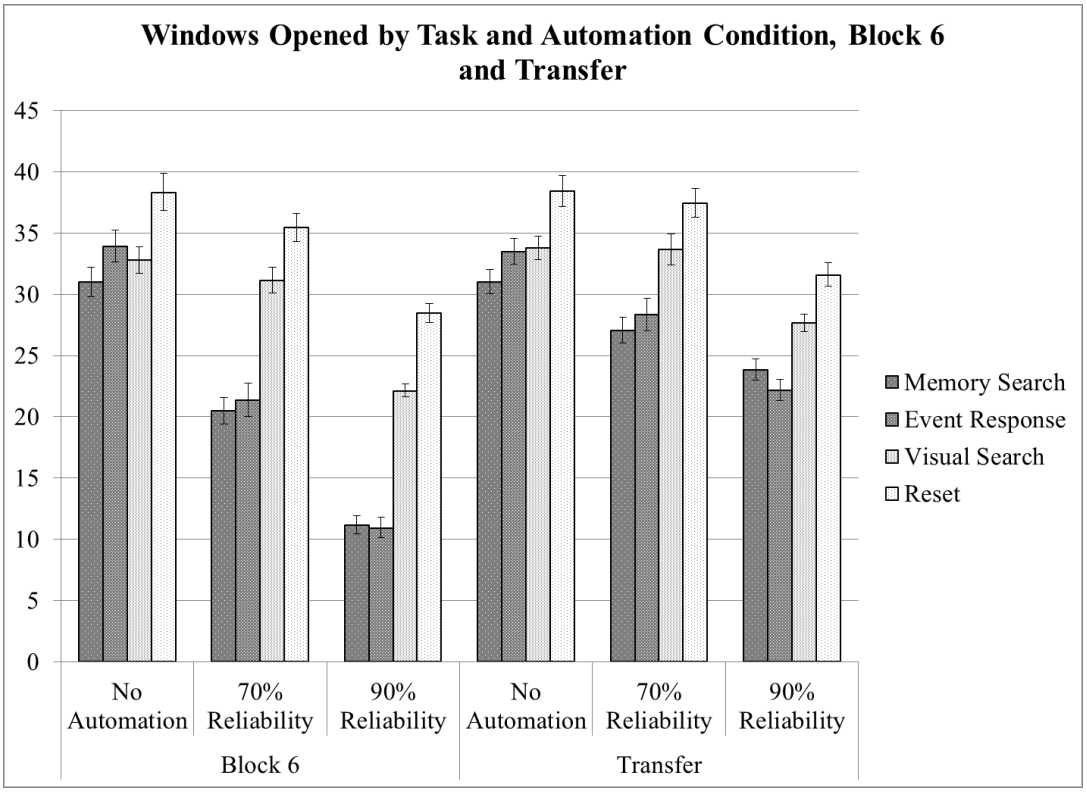

Figure 24. The number of windows opened by task and automation condition in block 6 and the transfer block with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

Table 16. Pairwise comparisons on windows opened for each task and automation condition between block 6 and transfer. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	P	T	p	t	p
Memory Search	0.04	0.967	-8.72	< .001	-15.45	< .001
Event Response	0.52	0.608	-9.41	< .001	-14.99	< .001
Visual Search	-1.10	0.276	-3.65	< .001	-7.76	< .001
Reset	-0.06	0.956	-2.85	0.006	-3.62	0.001

The number of windows opened in block 6 versus transfer is shown in Figure 24. The results of the pairwise t-tests are shown below in Table 16. Based in the results in the table, the no automation condition showed no differences between block 6 and the transfer block for any of the tasks. The 70% and 90% reliable conditions showed significantly more windows opened for all four tasks. The proportion difference is shown in Figure 25.

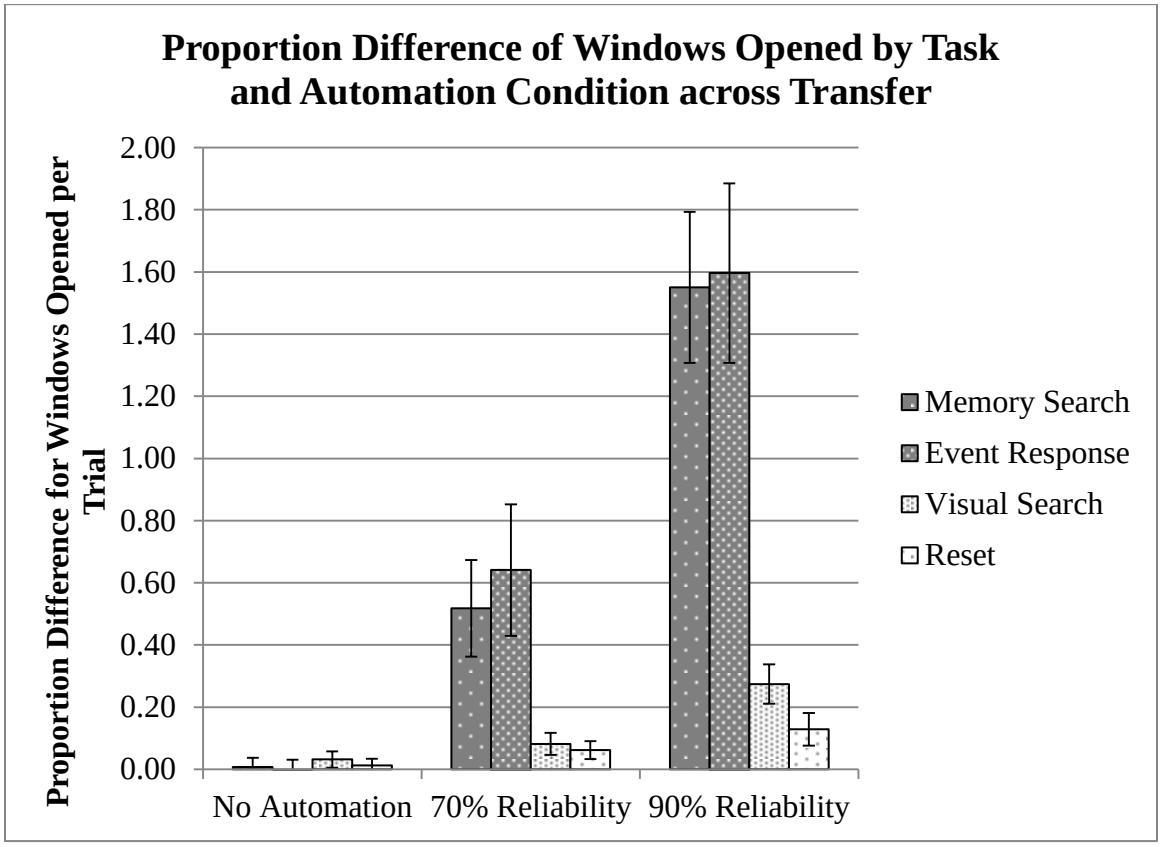

Figure 25. The proportion difference of windows opened by task and automation condition between block 6 and the transfer block with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

All four tasks showed significant increases in the two automated conditions, but the 70% reliable condition opened the two high-criticality/low-frequency tasks 50-60% more (as compared to ~10% more for the other two), while the 90% condition opened them 160-170% more (as compared to 15-25%).

To test whether the different tasks changed more or less for each condition as well as across conditions, I conducted two additional sets of pairwise t-tests, one comparing the four tasks against each other for each of the two automated conditions (Table 17), and one comparing the two automated conditions against each other for each task.

Table 17. Pairwise comparisons on proportion difference of windows opened between the different tasks for each of the different automation conditions in the transfer block. Shaded p values represent significant effects.

	70% Reliability		90% Reliability	
	t	p	t	p
Memory Search vs. Event Response	-2.00	0.048	-0.80	0.427
Memory Search vs. Visual Search	6.10	< .001	10.69	< .001
Memory Search vs. Reset	5.34	< .001	11.54	< .001
Event Response vs. Visual Search	5.42	< .001	8.72	< .001
Event Response vs. Reset	5.66	< .001	9.54	< .001
Visual Search vs. Reset	1.01	0.318	5.23	< .001

Based on the results in Table 17, the change for the high-criticality/low-frequency tasks was significantly higher than the low-criticality/low-frequency tasks for participants in both the 70% and 90% conditions, even though all were significantly above zero (differences from Table 16). Furthermore, the effect was graded, with the change being greater in the 90% than in the 70% in the two high-criticality/low-frequency tasks and the same in the other two. ($t = -3.58$, $p = .001$ for memory search; $t = -2.67$, $p = .011$ for

event response; t= -2.65, p = .013 for visual search; t = -1.12, p = .273 for reset). This may be due to the fact that both conditions began to interact with the system like the non-automated condition and that the 90% reliable condition had farther to shift to reach that level.

Points Scored

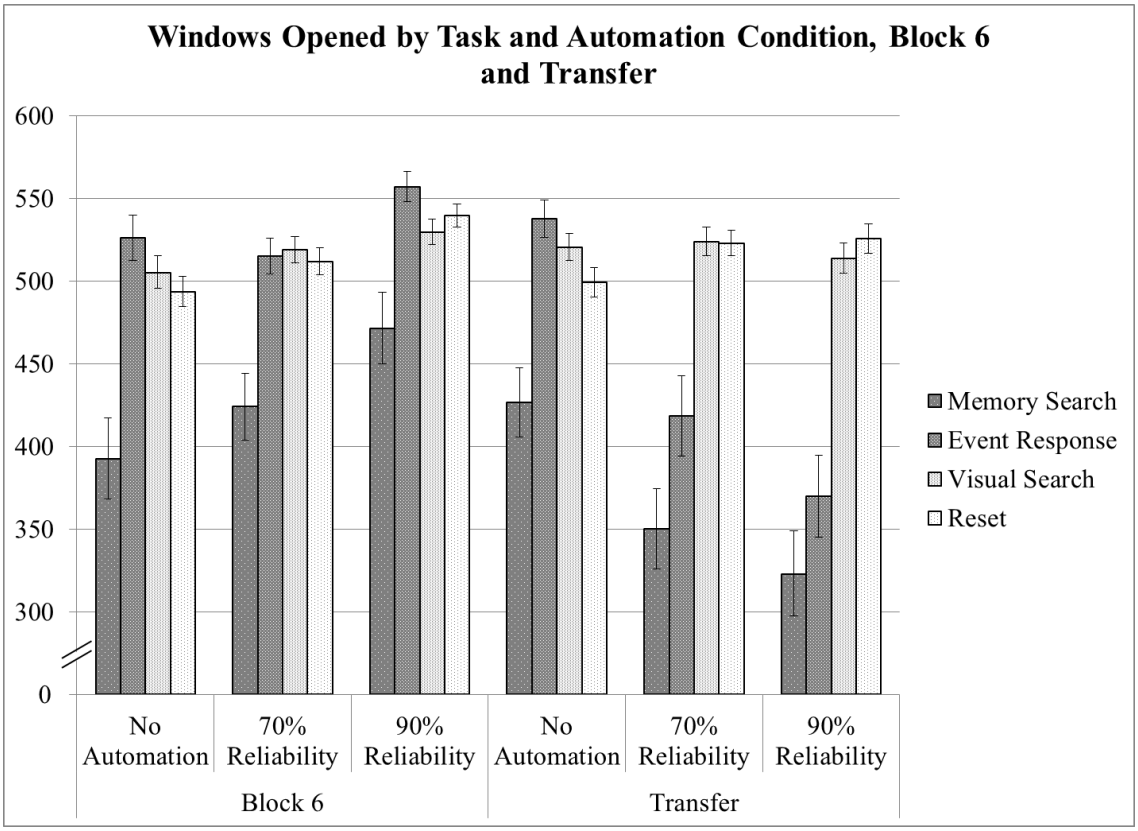

Figure 26. The number of points scored by task and automation condition in block 6 and the transfer block with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right. The maximum score for any of the tasks is 600.

Table 18. Pairwise comparisons on points scored for each task and automation condition between block 6 and transfer. Shaded p values represent significant effects.

	No Automation		70% Reliability		90% Reliability	
	t	p	t	p	t	p
Memory Search	1.55	0.125	3.21	0.002	5.33	< .001
Event Response	-0.70	0.485	4.56	< .001	6.83	< .001
Visual Search	-1.47	0.147	-0.52	0.605	1.88	0.064
Reset	-0.43	0.668	-1.43	0.158	1.39	0.169

The number of points scored in block 6 versus transfer is shown in Figure 26. The results of the pairwise paired t-tests are shown in Table 18. Based on these results, the no automation condition again showed no difference between block 6 and transfer, as expected. As for the automated conditions, both (70% and 90%) performed significantly worse on the two high-criticality/low-frequency tasks (memory search and event response) while not changing significantly on the other two tasks. This led to the conclusion that the significant point change between block 6 and transfer was due to those two tasks. Figure 27 shows the proportion difference scores for the three conditions.

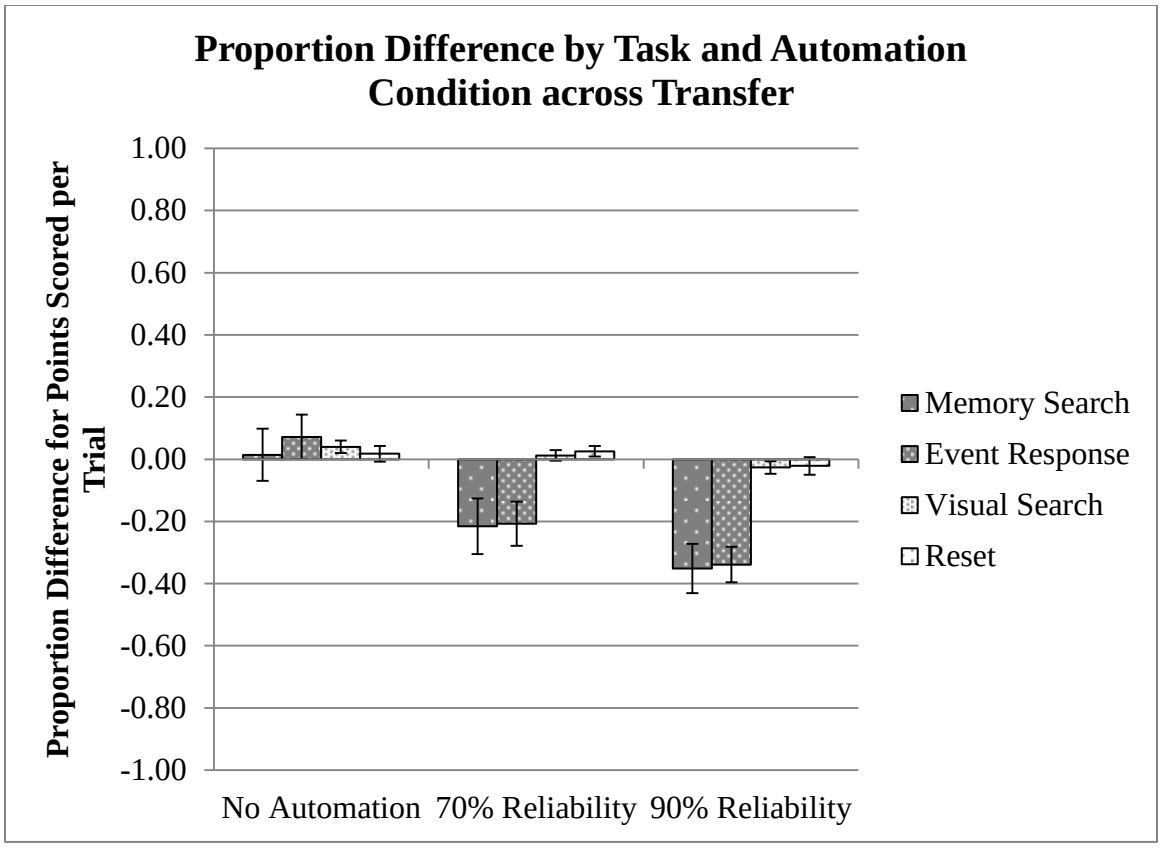

Figure 27. The proportion difference of points scored by task and automation condition between block 6 and the transfer block with standard error bars. Within each condition cluster, the tasks are grouped by their attributes: the two high-criticality/low-frequency tasks are on the left, while the two low-criticality/high-frequency tasks are on the right.

Again, no differences were found in the non-automated condition, or in the two low-criticality/high-frequency tasks (visual search and reset) for the automated conditions. The participants in the 70% reliability condition lost 15-20% of the points they gained in transfer on the high-criticality/low-frequency tasks, while 90% reliability condition participants lost 25-30% of their points on those tasks.

Like with windows, I conducted two additional sets of pairwise t-tests, one comparing the four tasks against each other for each of the two automated conditions

(Table 19), and one comparing the two automated conditions against each other for each task.

Table 19. Pairwise comparisons on proportion difference of points scored between the different tasks for each of the different automation conditions in the transfer block. Shaded p values represent significant effects.

	70% Reliability		90% Reliability	
	t	p	t	P
Memory Search vs. Event Response	0.34	0.737	0.61	0.543
Memory Search vs. Visual Search	-1.60	0.113	-3.33	0.001
Memory Search vs. Reset	-1.77	0.080	-3.74	< .001
Event Response vs. Visual Search	-4.46	< .001	-5.47	< .001
Event Response vs. Reset	-4.57	< .001	-5.54	< .001
Visual Search vs. Reset	0.81	0.685	-0.39	0.696

In the 70% condition, the event response task changed the most, followed by the other three. In the 90% condition, the two high-criticality/low-frequency tasks changed the most, followed by the other two. There were also no differences between the increases in the 70% and the 90% ($t = 1.14$, $p = .263$ for memory search; $t = 1.44$, $p = .159$ for event response; $t = 1.48$, $p = .146$ for visual search; $t = 1.42$, $p = .163$ for reset).

Summary

The workload of the participants in the previously automated conditions increased from the loss of automation, which caused them to open significantly more windows across the board. The largest differences in the amount of windows opened were to be found in the high-criticality/low-frequency tasks.

This is especially important in the context of the points scored; as the only significant differences in the score between block 6 and transfer were that the two

automated conditions gained fewer points in the tasks they opened proportionally more windows for.

Transfer Summary

The overall increase in workload and decrease in performance for the participants in the two previously automated conditions shows that they were indeed negatively affected by transfer. The taskwise analysis shows, however, that these negative effects were not uniform, those in the previously automated conditions suffered greater decreases in workload from the two high-criticality/low-frequency tasks, as well as the only significant points drops between block 6 and transfer. Their attention allocation strategies from the experimental trials were reliant on the automation, and the loss of that automation affected the way they interacted with the system. Across all measures and tasks, the originally non-automated condition showed no changes.

CHAPTER 4 – DISCUSSION

Key Findings

I found that the drop in workload provided by diagnostic automation came with a shift of attention allocation between the different tasks. The participants in the automated conditions opened the different tasks in a manner that followed their frequency, with the low-frequency tasks being opened less than the high-frequency ones. Because the low-frequency tasks were worth more points per event, this translated to a higher efficiency on those tasks. The effect of automation reliability was that of magnitude, with the efficiency gains of 70% automation being significant over no automation and 90% being significant over both.

Those tasks that had the highest gains with automation, however, were the ones most affected when the automation was taken away. Whereas those that had become practiced with the automation increased the number of times they opened all the tasks' windows, the magnitude of the change due to transfer was much greater for the high-criticality/low-frequency tasks. Furthermore, the only significant point changes in transfer were found in those tasks. Again, the effect was graded, with those participants in the higher reliability condition losing more points and opening relatively more windows in transfer.

Overall Effect of Automation

This study was designed to determine what effect diagnostic automation at differing levels of reliability had on visual attention allocation in a multiple-task environment. I looked at this in two ways: One, I studied the behavior of participants

interacting with different levels of automation reliability (as well as a lack of automation altogether) across a number of trials. Two, I removed the automated aid for all participants to determine what effect automation had by removing it, thereby simulating an automation failure.

More specifically, I wanted to assess the effects of that automation on attention allocation, as it had not previously been directly measured and had been identified as being an important first step in human information processing (the sensory processing stage of the Parasuraman, Sheridan, and Wickens (2000) model). To measure attention allocation, I used the metric of the number of windows opened across a trial. My judgment was that, while the participant had a window open, he or she could not visually attend to any of the other tasks. The metric would give a measure of the number of times they allocated their attention to each task over a trial. It is, however, not the only possible implicit measure of attention allocation. Other measures are possible (eye-tracking, self-reporting, etc.), but this method best fit the scope and goal of the current study.

The automation I chose was diagnostic automation, that is, automation that is designed to aid attention allocation by providing alarms and alerts to important parts of the system. Diagnostic automation was chosen because it is intended to decrease workload by helping the operator know when and where to look. This, in turn, reduces activation and switching costs (Altmann & Trafton, 2002; Wickens & McCarley, 2008) by removing the need to decide what task to activate and when to switch. The current study supports this idea, as automation had a graded benefit to workload, with automation helping overall and highly reliable automation helping more.

Automation also purports to aid performance at sufficiently high levels of reliability (Wickens & Dixon, 2007). In this study, the highly reliable automation did significantly aid performance but the lower automation had mixed effects, showing an overall benefit by starting out significantly better but performing similarly to no automation at the end. This does not follow the results from Wickens and Dixon (2007), as the lower automation provided some benefit over no automation (at certain times), but the reasons for this are not readily apparent from my data. Further investigation may be required.

Furthermore, both types of automation helped more at the beginning of the trials, giving the participants in those conditions the largest benefit early. This could show a great benefit to training with an automated system, as it decreases the amount of time needed to have the participants reach a stable performance level. Again, as this was outside the scope of the current experiment, more investigation is warranted to discover how participants learn under different levels of reliability.

It should be noted that, as automation affects workload and performance differently, the two cannot be seen as one; a decrease in workload may not mean a gain in performance. This was especially apparent in the behavior of the participants in the 70% reliable condition in the current study, as they opened significantly fewer windows by the end but performed at a similar level as those with no automation.

In practice, this means that workload and performance must be measured independently and automation designed with a high enough level of reliability to support and improve both.

Another way of measuring workload and performance side by side is to calculate some measure of efficiency, as was done in the current study. I found that, overall, as any automation benefitted workload and highly reliable automation benefitted performance, automation overall benefitted efficiency, with the highly reliable automation benefitting it more.

Allocation Strategies across Tasks

The main purpose of this study was to determine how automation reliability affected the attention allocation of the participants interacting with the system, as no studies had researched how reliability affected more than two tasks. To this end, I created system parameters that were designed to show the differences between conditions. More specifically, I varied two task attributes, frequency and criticality. Frequency was varied due to the possibility that automation would provide more reinforcement to open the critical tasks (Herrnstein, 1961). Criticality was varied as a measure of the quality of feedback (Baum, 1974), a way to balance the bias all participants would have toward the more frequent tasks. I thought the added feedback given by the automation (to lead the participants to the correct windows at the correct times) might bias those participants toward a frequency-driven allocation strategy.

The results of the study show just that; a bias in the automated conditions away from the low-frequency tasks. The highly reliable automation supported this strategy throughout the experiment, whereas the less reliable automation provided for a similar strategy by the end. The performance showed that, regardless of the bias towards frequency, those in the automated conditions were performing as well or better than their non-automated counterparts in the high-criticality/low-frequency task types.

Because there was an interest in this study to make sure no task was any more important overall than the others, these two attributes had to be confounded to keep the same overall point value constant for each task (as points per task went up, the number of times the task happened had to go down, and vice versa). Further studies might see how the current effects are changed when these two are varied independently.

Furthermore, the criticality and frequency were allocated to certain tasks based on how they affected the difficulty of the tasks and the task space overall. Further studies might reallocate the tasks to different frequency and criticality groups to see how much the specific task attributes not varied or measured were responsible for the results found in this study.

The tasks themselves also made a difference, as certain tasks were checked relatively more (the reset task) or performed worse (The memory search task in the no automation and 70% conditions) regardless of automation. This may be due to other inherent task attributes; the reinforcement of continual information presented by the task (Herrnstein, 1961) or the inherent difficulty of the task itself. The STEP system was made up of different types of tasks to better simulate a working environment, one where many different demands are placed on an operator and the operator must continue to switch between them to keep the overall system running well. Further studies might be done with four of the same task to avoid these concerns.

Based on the study, then, the amount that diagnostic automation supports a set of tasks depends on the characteristics of those tasks, that is, diagnostic automation supports some types of tasks better than others. Designing an automation to support a multiple task environment requires analyzing the tasks to be automated and deciding which ones

would most benefit from that automation. Based on the results from this study, tasks that are critical to operation yet do not happen often would benefit most, as the alert would call attention to them when needed and allow the operator to attend to other tasks when the alert was silent.

Loss of Automation

As stated before, the loss of automation often has drastic effects on previously automated tasks (Ma & Kaber, 2007). It also provides a measure of how much the participants in the automated conditions were relying on that automation; the more loss caused when the automation is taken away, the more the participant needed the automation to perform at the level at which they had been performing.

In this study, the loss of automation caused those using automation at any level to suffer, both because of increased workload and decreased point score. Furthermore, the higher the level of reliability of the aid, the more it disrupted them, meaning that good automation may not be the best option when it fails, as the operator may not be able to perform at an acceptable level without the usual aid. These effects were found even though the participants in the current study were told to expect the change; in the operational environment, operators may not receive such alerts prior to an automation failure.

The greatest workload and performance drops, however, were had in the high-criticality, low-frequency tasks. All automated participants opened more windows in every task, but only suffered a point loss in those two tasks. Furthermore, the magnitude of the change in both the windows opened and the points lost were much higher in those

tasks, with those with highly reliable automation rising higher in windows and dropping further in points than those with the less reliable.

The automation of these tasks caused them to suffer the most when that automation was taken away. This effect should also be taken into account, as losing performance on critical tasks due to automation failure could easily have major consequences in certain fields and situations. What is most important is that the type and attributes of each task should be brought into consideration when designing automated systems and training operators to use them.

Next Steps

The analysis that might most inform the current study is one of strategy, that is, the strategies that the participants reported using to decide which windows to open and when to respond to or ignore the warnings of the automated alerts. As it stands currently, the reasons as to why, for example, those in the automated conditions opened fewer windows in the high-criticality/low-frequency tasks are only based upon previous literature and inference; a better understanding of *why* the participants are doing what they are doing would allow us to learn the underlying processes behind these action patters, as well as aid in the design of training and support materials.

Analysis of learning effects and how automation affects training would also be a good way to proceed, as the current study was not designed to see how automation affects the early stages of training or how much about the system the participants learned with and without automation. The effects behind the higher performance and lower workload seen early in the automated conditions need to be better understood, as they may explain

more how automation affects learning of a system. This is important both in terms of the behaviors involved, but also for the effects on training.

Future research efforts will be important to more fully understand the potential benefits of automation as well as attention allocation strategies. The present study provides valuable insights into the important role of automation for people monitoring multiple systems. Reliable automation can guide attention resulting in less workload and higher performance. This in turn, makes for a more efficient interaction. However, there is a cost when the automation fails: the effect the automation has on behavior persists after that automation is taken away, causing the subsequent system use to become inefficient.

APPENDIX A – MEMORY SEARCH TASK

The memory search task was a modified Sternberg memory scanning task, adapted from Sternberg (1969) and Fisk and Rogers (1991). A similar task was in the original SYNWORK1 (Elsmore, 1994) system, as well. It was chosen due to address the common cognitive demands of working memory, demands used by any task that requires the operator to keep certain values in mind for use later.

Memory scanning tasks require the creation of a stimulus ensemble, a series of stimuli from which all positive sets and test stimuli are determined. Based on previous studies, the stimulus ensemble for this task was the letters A, C, D, E, M, R, S, U, and Z (Fisk & Rogers, 1991). The task flow (shown in Figure 28) was as follows:

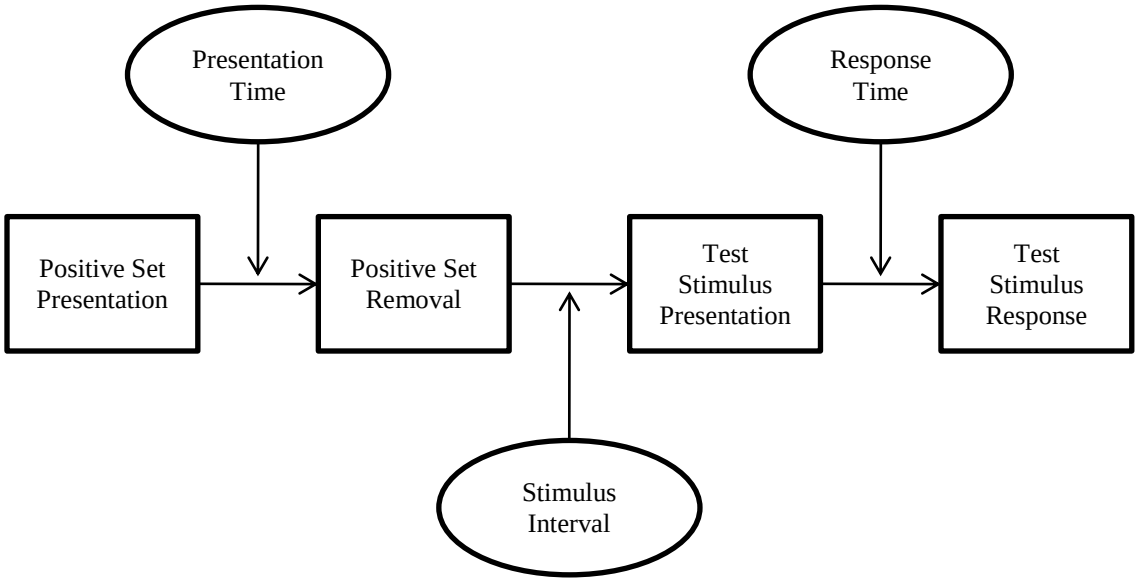

Figure 28. The task flow for the memory search task.

First, the system generated a positive set by selecting exactly six letters from the stimulus ensemble above. This is different from the original Sternberg task in which varying set sizes were selected. The specific positive set size in the current study was chosen to dictate a certain level of difficulty of the memory demand.

This positive set was then presented to the participant, as illustrated in Figure 29. The positive set stayed visible for a minimum of 10 seconds or until the participant pressed the "CONTINUE" button or closed that task's window, whichever came first.

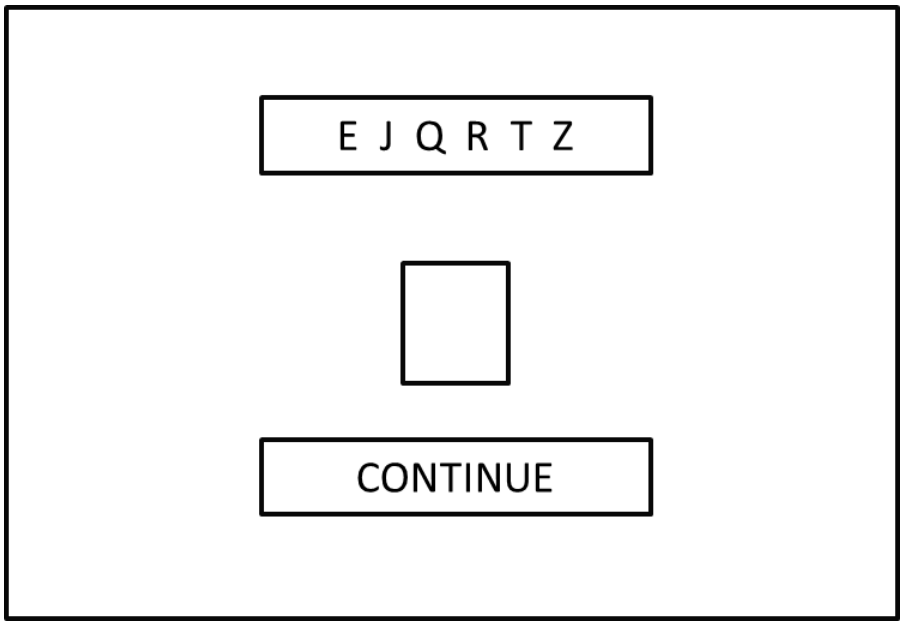

Figure 29. The memory set presented to the participant.

After the positive set was taken away, the task space remained blank for a certain period of time designated by the system. This period of time was generated to be between 50 and 70 seconds, for an average of 5 probes a trial.

When the interim time was over, the system generated a test stimulus. This test stimulus was either from the positive set or from the other letters in the stimulus

ensemble, labeled by Sternberg (1969) as the negative set. The probability that the stimulus was from either set was 50%. Once this test stimulus was generated, it was presented to the participant, as shown in Figure 30.

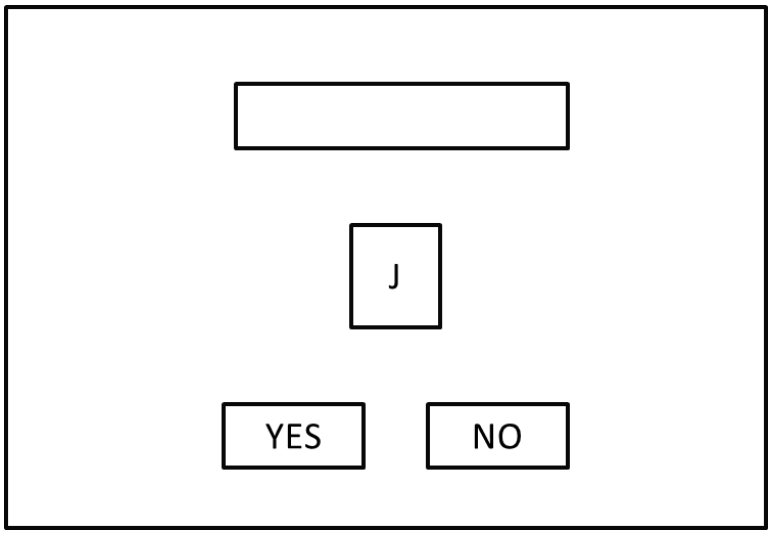

Figure 30. The test stimulus presented to the participant.

Participants were instructed to press the YES button if the letter was from the positive set and NO if from the negative one. They were scored based on whether or not they answered correctly. Correct answers were rewarded 120 points. Incorrect answers had 60 points deducted. The participant had 10 seconds to answer, after which the lack of an answer was counted as incorrect. When the user responded or the task timed out, the task started over by presenting a new memory set like the one in Figure 29.

The automation in this task was a thick red border around the outside of the task at the time of the test stimulus presentation. This aided participants by alerting them of the exact time of the test stimulus presentation. This red border appeared regardless of

which task window was open and lasted 10 seconds or until the task's window was opened before disappearing.

The automation made two types of errors: misses, defined as when the test stimulus appeared and the red border did not appear, and false alarms, defined as when, between the removal of the positive set and the presentation of the test stimulus, the red border appeared without coinciding with the appearance of the test stimulus. The false alarms appeared for 10 seconds.

APPENDIX B – VISUAL SEARCH TASK

The visual search task was adapted from McBride, Rogers, and Fisk (2010). It was chosen to address the cognitive demands of visual search, common to tasks that require the operator to find specific pieces of information while providing many different types. The task, when shown, always looked the way it appears below in Figure 31.

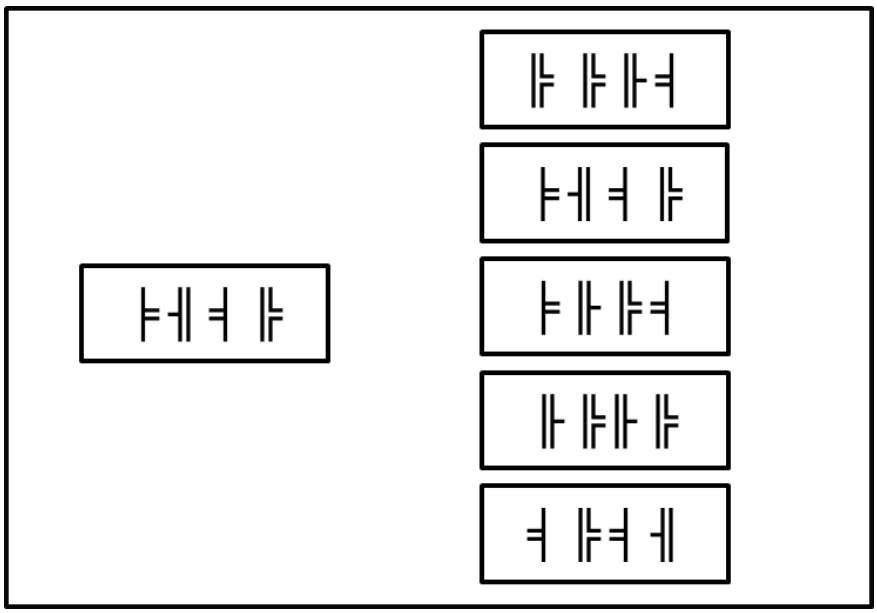

Figure 31. The visual search task.

The set (group of barcodes) of characters shown in Figure 31 on the left side was the target set. The boxes on the right showed the possible matching sets. The task of the participant was to find the set on the right that matches the target set. A visual search such as this one appeared many times throughout the trial randomly between 10 and 20 seconds, with 20 sets being presented every trial. The participant had until the next search appears to answer the search. They were scored based on whether or not they

answered correctly. Correct answers were rewarded 30 points. Incorrect answers had 15 points deducted. After the participant responded, all the boxes went blank until the next set was presented.

The automation in this task was a thick red border around the task space when the target and matching sets appeared. This aided participants by alerting them of the exact time of the sets appearing. This red border appeared regardless of which task window was open and lasted 10 seconds or until the task's window was opened before disappearing.

The automation made two types of errors: misses, defined as when the sets appeared and the red border did not appear, and false alarms, defined as when the sets did not appear and the red border did. The false alarms appeared for 10 seconds.

APPENDIX C – RESET TASK

The reset task was adapted from SYNWORK1 (Elsmore, 1994, Sit & Fisk, 1999). This reset task was made to operate like a car tachometer; it addressed the cognitive demands of monitoring, requiring the participant to monitor the bar until it reached an optimum level and then react before it left that level. The reset task always appeared similar to the way it does below in Figure 32.

As the beginning of the task, the top bar started at the left gauge mark, the one demarked by being longer than the others. The bar then moved steadily toward the right end. The time taken to move from the left to right was defined as being between 10 and 20 seconds. The bar moved from one side to the other 20 times in each trial. When the bar reached the end of the gauge, it reset to the beginning.

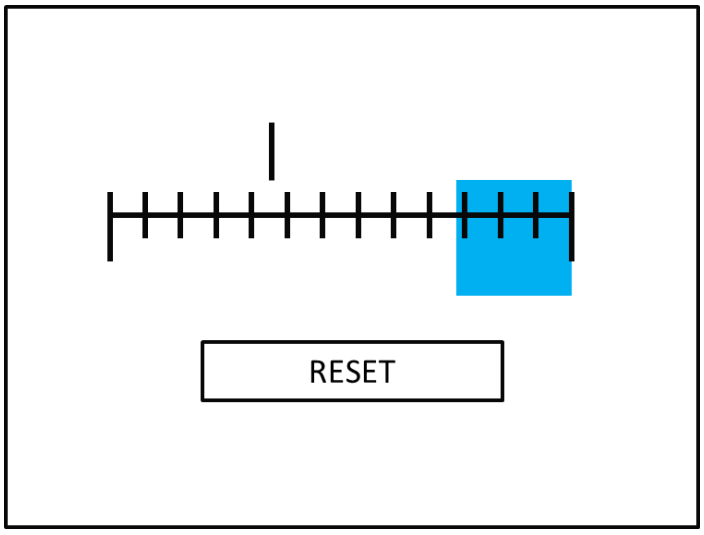

Figure 32. The reset task.

The task of the participant was to reset the top bar before it reached the right end. For each bar passed, the participant gained points up to the blue area at the rate of 3 points per gauge bar for resetting the bar. While the top bar was within the blue area but not at the end, maximum points (30) were awarded for resetting the gauge. When the bar reached the end, however, 15 points were deducted. The participant reset the bar by pressing the RESET button. This also caused the top bar to reset to the left.

The automation in this task was a thick red border around the task space when the indicator bar started moving. This automation aided participants by alerting them that points were possible and that the task would soon deduct points if not attended to. The red border appeared regardless of which task window was open and disappeared 10 seconds after appearing or when the task's window was opened.

The automation made two types of errors: misses, defined as when the top bar started moving without the appearance of the red border, and false alarms, defined as when the red border appeared sometime before the top bar started moving. The false alarms appeared for 10 seconds.

APPENDIX D – EVENT RESPONSE TASK

The event response task was adapted from SYNWORK1 (Elsmore, 1994, Sit & Fisk, 1999). This task was designed to address the cognitive demands of monitoring and stimulus response. It operated like many computer alerts: the alert popped up and the user had a certain amount of time to respond before the alert goes away.

The event response task has two states. Normally, the task was in an "event negative" state, where nothing is required of the participant. In this normal state, the task looked like Figure 33.

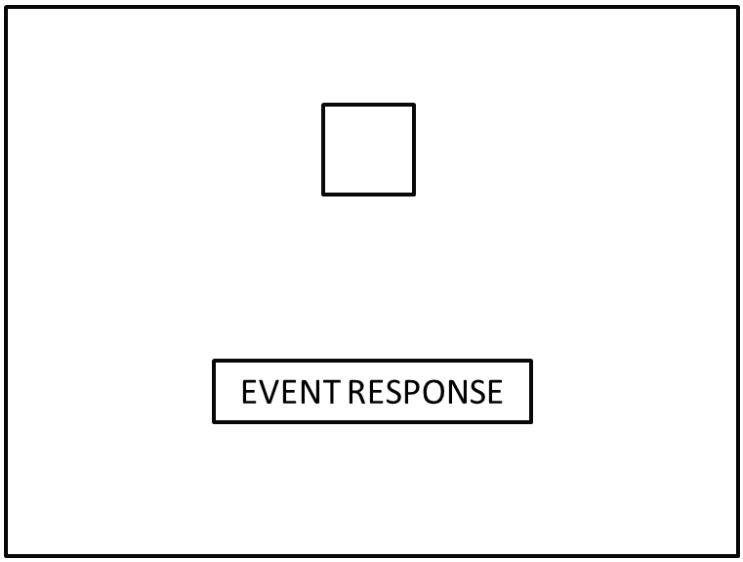

Figure 33. The "event negative" state of the event response task.

After a number of seconds selected by the system to be between 50 and 70, the task went into an "event positive" state. This happened five times per trial. This event positive state required that the participant respond by pressing the EVENT RESPONSE button within 10 seconds. If the button was pressed in time, 120 points were rewarded.

If not, 60 points were deducted. The event positive state was denoted by a change in the upper box, which turned from a white fill to a green fill, shown in Figure 34. Sixty points were also deducted if the participant pressed the EVENT RESPONSE button when the event had not happened (when the box was white).

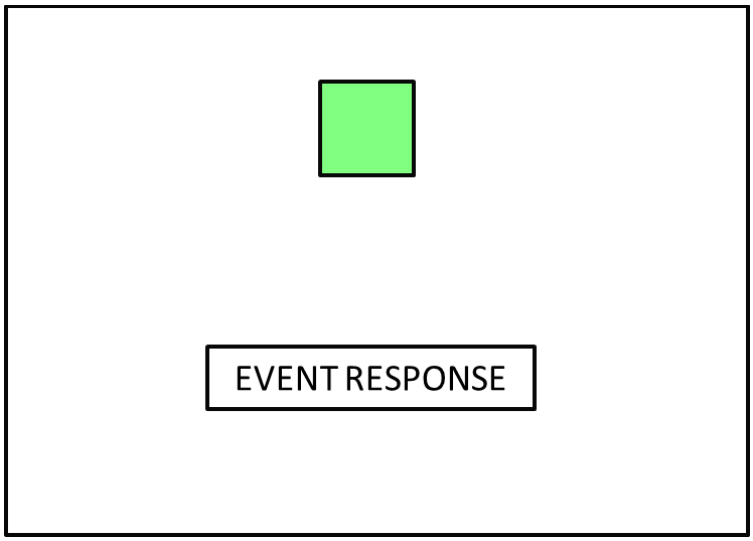

Figure 34. The "event positive" state of the event response task.

The automation in this task was a thick red border around the task space when the task state became "event positive". This automation aided participants by alerting them that the event had happened and needed a response. The red border appeared regardless of which task window was open and disappeared 10 seconds after appearing or when the task's window was opened.

The automation made two types of errors: misses, defined as when the task state changed to "event positive" without the appearance of the red border, and false alarms, defined as when the red border appeared without the task state changing. The false alarms appeared for 10 seconds.

APPENDIX E – EXPERIMENTAL PROTOCOL

Day 1		**Day 2**	
Task	**Time (min)**	**Task**	**Time (min)**
Informed Consent	5	Refresher Trial	5
Demographics and Health Questionnaire	12	**Break**	2
Break	**5**	Experimental Block 4	20
Isolated Practice	12	**Break**	**2 – 5**
Combined Practice	6	Experimental Block 5	20
Break	**5**	**Break**	**2 – 5**
Experimental Block 1	20	Experimental Block 6	20
Break	**2 – 5**	Reverse Digit Span	5
Experimental Block 2	20	Digit Symbol Substitution	5
Break	**2 – 5**	**Break**	2-5
Experimental Block 3	20	Transfer Block	20
Break	**2 – 5**	**Break**	**2-5**
Vocabulary Test	6	Strategy Questionnaire	10
Visual Acuity Test	2	Debriefing	3
Total Time	119-125	Total Time	138-144

APPENDIX F – CONSENT FORM

Georgia Institute of Technology

Project Title: Attention Allocation and Automation in a Multiple-Task Environment

Investigators: Dr. Arthur D. Fisk & Dr. Wendy A. Rogers (Principal Investigators)
Ralph Cullen (Student Investigator)

Protocol and Consent Title: Main 09/18/10v1

Purpose:
You are being asked to be a volunteer in a research study. The purpose of this form is to tell you about the tasks you will be asked to complete today and to inform you about your rights as a research volunteer. Feel free to ask any questions that you may have about the research study and what you will be asked to do.

Thank you for your interest in participating in this research study. Our work could not be completed without the help of volunteers. The purpose of our research is to determine the effects of automation reliability on attention allocation strategies in multiple-task environments; that is, when and where people look when asked to do multiple tasks at once. We expect to enroll 84 people age 18-28 in this research study.

Procedures:
If you decide to be in this study, your part will involve taking a number of general tests that measure your abilities, including vocabulary, memory, spatial ability, perceptual speed, and vision.

When you arrive you will be asked to complete several questionnaires to collect general demographic and health information. Next, you will be asked to complete several tasks on the computer. Sometimes you will be performing one task at a time and sometimes you will be performing several tasks at once or at the same time. You will be given full instructions on how to do each task. You will be randomly assigned to one of four groups that differ by the amount and reliability of automation provided. Automation refers to the aid you may be given in completing the tasks. All tasks will be done individually.

Some of the tasks you will complete will be performed on a computer. To ensure that your responses will be accurately documented, your computer screen will be recorded.

Remember that you will be given full instructions on every task. It is important that everyone understands the instructions before beginning the tasks. Because we are trying to measure a range of abilities, some of the tasks are very simple, and others are quite difficult. If anything is unclear at any time, please do not hesitate to ask questions.

This 2-day study will take approximately 6 hours of your time (approximately 3 hours per day over two consecutive days). You may stop at any time and for any reason. Breaks will be provided throughout the study.

Risks/Discomforts

The following risks/discomforts may occur as a result of your participation in this study: Participation in this study involves minimal risk or discomfort to you. Risks are minimal and do not exceed those of normal office work. Please tell us if you are having trouble with any task.

Benefits

You are not likely to benefit in any way from joining this study. We hope that others will benefit from what we find in doing this study.

Compensation to You

You will receive 1 Experimetrix credit for each hour of participation. The time to complete the study is approximately 6 hours, so you will receive 6 Experimetrix credits if you complete the study. If you withdraw from the study early for any reason, you will receive 1 credit per hour for your time.

Confidentiality

The following procedures will be followed to keep your personal information confidential in this study: All written data that are collected about you will be kept private to the extent allowed by law. To protect your privacy, your written records will be kept under a code number rather than by name. Your written records will be kept for archival purposes and stored in locked files that only study staff will have access to. Your name and any other fact that might point to you will not appear when results of this study are presented or published.

Your privacy will protected to the extent allowed by law; your personal information may be disclosed if required by law. This means that there may be rare situations that require us to release personal information about you, for example, in case a judge requires such release in a lawsuit or if you tell us of your intent to harm yourself or others (including reporting behaviors consistent with child abuse).

To make sure that this research is being carried out in the proper way, the Georgia Institute of Technology IRB will review study records. The Office of Human Research Protections may also look at study records.

Because each individual's data and test scores are completely confidential, we cannot mail your individual results.

Costs to You

There are no costs to you associated with participating in this study.

In Case of Injury/Harm

If you are injured as a result of being in this study, please contact Dr. Wendy A. Rogers at (404) 894-6775 or Dr. Arthur D. Fisk at (404) 894-6066. Neither the Georgia Institute of Technology nor the principal investigators have made provision for payment of costs associated with any injury resulting from participation in this study.

Participant Rights

- Your participation in this study is voluntary. You do not have to be in this study if you do not want to be.
- You have the right to change your mind and leave the study at any time without giving any reason and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.
- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

Questions about the Study or Your Rights as a Research Participant

- If you have any questions about the study, you may contact the investigator obtaining consent (listed below) at (404) 894-8344.
- If you have any questions about your rights as a research participant, you may contact Ms. Kelly Winn, Georgia Institute of Technology, Office of Research Compliance, at (404) 385-2175 or kelly.winn@gtrc.gatech.edu.

If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.

_____

Participant Name (please print)

Participant Signature **Date**

_____

Name of Investigator Obtaining Consent (please print)

Signature of Investigator Obtaining Consent **Date**

APPENDIX G – STRATEGY QUESTIONNAIRE

Subjective Experience Questionnaire

Now that you have completed our experiment, we would like you to answer a few questions about your experience in the study. There are no right or wrong answers, please just provide your opinion.

1) Overall, how often did/do you think the automated system correctly alerted you to complete a task that needed attention (0-100%)?

_____%

2) For each task, how often did/do you think the automated system correctly alerted you to complete a task that needed attention (0-100%)?

Task 1 (0-100%) _____	Task 2 (0-100%) _____
Task 3 (0-100%) _____	Task 4 (0-100%) _____

3) Overall, how much did you trust the automated system to correctly alert you (Please circle your answer)?

1 2 3 4 5

Very Little Neutral Very Much

4) For each task, how much did you <u>trust</u> the automated system to correctly alert you (Please circle your answer, 1 = Very Little, 3 = Neutral, 5= Very Much)?

Task 1 (1-5) 1 2 3 4 5	Task 2 (1-5) 1 2 3 4 5
Task 3 (1-5) 1 2 3 4 5	Task 4 (1-5) 1 2 3 4 5

5) Please describe the approach (e.g. a strategy, trick, or technique) you used during Day 1 to find the tasks that currently needed responses.

_____

_____

_____

_____

6) Did this approach change on Day 2 (Please circle your answer)?

No Yes

If Yes, please explain how it changed.

_____

_____

_____

_____

APPENDIX H – DEBRIEFING FORM

Attention Allocation and Automation in a Multiple-Task Environment

Thank you very much for participating in this research study. We could not conduct our research without the help of volunteers like you.

The goal of this study is to learn how different levels of automation reliability influence the way people attend between tasks in a multiple-task environment. Multiple-task environments are all around us. For example, when driving a car, the driver must attend to the task of keeping the car on the road, the task of keeping the correct speed, the task of navigating to the correct destination, and any other tasks in the car such as changing the radio, or carrying on a conversation. These tasks can be very difficult to do all at the same time, so automations have been developed to help: things like cruise control for the speed and GPS for the navigation. These automations are not always perfect, however, and when automation is not perfect, performance can suffer.

In this study, we want to see how this imperfect automation help affects the way participants allocate their attention between tasks. Different participants had to operate systems that differed with respect to the amount and quality of automation help given:

- One group had no automation
- One group had automation that was right 100% of the time
- One group had automation that was right 70% of the time

In the first session and most of the second session, you completed many trials working in a multiple-task environment. The multiple-task environment included four tasks: memory search, continuous tracking, reset, and event response. We wanted to give you time to get comfortable with using the system and have the opportunity to develop a strategy. We expected to see differences in the strategies people used to allocate their attention based on whether or not automated help was available and how reliable that automated help was. For each task we measured: number of times accessed and points earned. The number of times accessed showed us which tasks you were looking at the most. The points earned gave us an idea of your overall performance.

At the end of the second session, we removed the automation for all participants. We wanted to see how participants' experience working with automation would influence the way they reacted and recovered when the automation was removed. Again, for each task we measured: number of times accessed and points earned.

It is important to learn how people allocate attention when they use automation because of the amount of time people spend working in multiple-task environments with imperfect automations. For example, if automation causes you to look more and spend more time on tasks you would otherwise ignore or only glance at occasionally, it might change the number of errors you make on those ignored tasks or to your performance overall, even if you do better on the task the automation is helping.

Furthermore, understanding how people respond when the automation is taken away will help us better predict what will happen when systems fail and the operators have to recover.

These results could also inform the design of products that have automated help. If automation and its reliability impact the way we allocate our attention when performing tasks, product developers may modify their approach to system design.

Thank you for your time and cooperation.

Student Investigator
Ralph H. Cullen (404) 894-8344

Principal Investigators
Dr. Arthur D. Fisk (404) 894-6066
Dr. Wendy A. Rogers (404) 894-6775

Human Factors and Aging Laboratory
Georgia Institute of Technology
http://www.hfaging.org/

APPENDIX I – TRIAL, BLOCK, AND DAY RESULT SUMMARY

Table 20. Mixed ANOVA data for the Trial, Block, and Day levels for the three dependent measures in the study.

BY TRIAL	Windows				Points				Efficiency			
	F	P	Power	partial η^2	F	p	Power	partial η^2	F	p	Power	partial η^2
Automation Condition (AC)	11.98	< 0.001	0.99	0.30	8.98	< 0.001	0.97	0.24	36.50	< 0.001	1.00	0.57
Trial	5.08	< 0.001	0.97	0.09	5.34	< 0.001	1.00	0.09	6.82	< 0.001	1.00	0.11
Task	136.28	< 0.001	1.00	0.71	16.56	< 0.001	1.00	0.23	56.59	< 0.001	1.00	0.51
Trial * AC	2.34	< 0.001	0.88	0.08	1.47	0.105	0.89	0.05	3.17	< 0.001	0.99	0.10
Task * AC	21.49	< 0.001	1.00	0.43	1.26	0.286	1.00	0.23	23.04	< 0.001	1.00	0.46
Trial * Task	6.17	< 0.001	1.00	0.10	3.30	0.001	0.99	0.06	4.65	< 0.001	1.00	0.08
Trial * Task * AC	1.45	< 0.001	0.94	0.05	1.79	0.023	0.96	0.06	2.56	< 0.001	1.00	0.09
BY BLOCK	Windows				Points				Efficiency			
	F	P	Power	partial η^2	F	p	Power	partial η^2	F	p	Power	partial η^2
Automation Condition (AC)	50.50	< 0.001	1.00	0.30	33.52	< 0.001	1.00	0.22	152.61	< 0.001	1.00	0.90
Trial	17.69	< 0.001	1.00	0.07	14.26	< 0.001	1.00	0.06	28.41	< 0.001	1.00	0.11
Task	518.93	< 0.001	1.00	0.69	46.22	< 0.001	1.00	0.16	224.96	< 0.001	1.00	0.49
Trial * AC	7.63	< 0.001	1.00	0.06	2.76	0.005	0.94	0.02	12.88	< 0.001	1.00	0.10
Task * AC	81.33	< 0.001	1.00	0.41	3.87	0.003	0.93	0.03	91.26	< 0.001	1.00	0.44
Trial * Task	17.47	< 0.001	1.00	0.07	1.67	0.097	0.76	0.01	9.36	< 0.001	1.00	0.04
Trial * Task * AC	3.40	< 0.001	1.00	0.03	1.40	0.128	0.88	0.01	4.57	< 0.001	1.00	0.04
BY DAY	Windows				Points				Efficiency			
	F	P	Power	partial η^2	F	p	Power	partial η^2	F	p	Power	partial η^2
Automation Condition (AC)	132.02	< 0.001	1.00	0.27	68.26	< 0.001	1.00	0.16	404.77	< 0.001	1.00	0.53
Trial	33.59	< 0.001	1.00	0.05	15.70	< 0.001	0.98	0.02	116.51	< 0.001	1.00	0.14
Task	1206.92	< 0.001	1.00	0.63	70.96	< 0.001	1.00	0.09	539.02	< 0.001	1.00	0.43
Trial * AC	18.91	< 0.001	1.00	0.05	2.25	0.106	0.46	0.01	52.07	< 0.001	1.00	0.13
Task * AC	188.90	< 0.001	1.00	0.35	6.02	< 0.001	0.99	0.02	216.89	< 0.001	1.00	0.38
Trial * Task	43.05	< 0.001	1.00	0.06	3.01	0.042	0.63	0.00	35.34	< 0.001	1.00	0.05
Trial * Task * AC	8.54	< 0.001	1.00	0.02	1.46	0.203	0.50	0.00	16.82	< 0.001	1.00	0.05

Shading denotes significance at the $\alpha = .05$ level.

APPENDIX J – POINT SCORE CORRECTION

As stated in the method, the number of points scored for each task and for the trial overall was affected by the number of windows opened: for each window opened, two points were deducted from the participants score for that task. Because of this interaction with the point scores, the point measure analysis was done once using the raw scores (with the window point loss) and again with a set of corrected scores (with the window loss points added back in). Provided below is the formula for correcting the raw score.

$$Corrected\ Score = Raw\ Score + 2 * Number\ of\ Windows\ Opened$$

Both analyses produced similar results, so all analyses reported in this paper use the corrected point score to avoid the interaction. Figure 35 illustrates the difference.

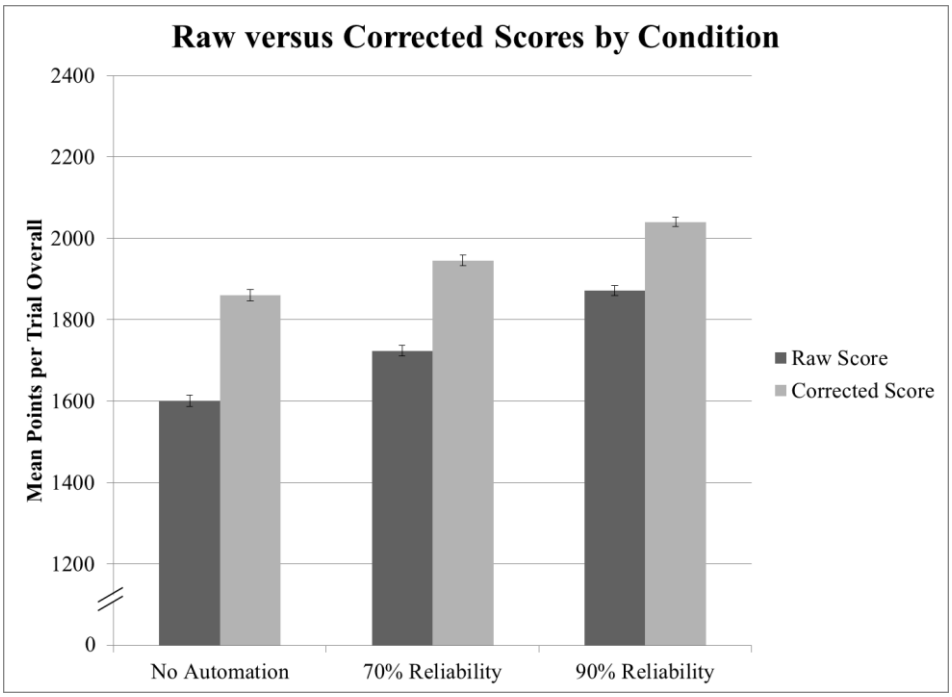

Figure 35. An example of the score correction, showing the main effect of automation condition.

REFERENCES

Altmann, E. M. & Trafton, J. G. (2002). Memory for goals: An activation-based model. *Cognitive Science, 26,* 39-83.

Baum, W. M. (1974). On two types of deviation from the matching law: Bias and undermatching. *Journal of the Experimental Analysis of Behavior, 22,* 231-242.

Bliss, J. P. & Acton, S. A. (2003). Alarm mistrust in automobiles: How collision alarm reliability affects driving. *Applied Ergonomics, 34,* 499-509.

Bliss, J. P. & Dunn, M. C. (2000). Behavioural implications of alarm mistrust as a function of task workload. *Ergonomics, 43*(9), 1283-1300.

Breznitz, S. (1984). *Cry wolf: The psychology of false alarms.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Cullen, R. H, Dan, C. S., Arivazhagan, R., & Rogers, W. A. (2011). Simultaneous task environment platform (STEP): Beta manual (HFA-TR-1106). Atlanta, GA: Georgia Institute of Technology, School of Psychology, Human Factors and Aging Laboratory.

Czaja, S.J., Charness, N., Dijkstra, K., Fisk, A.D., Rogers, W.A., & Sharit, J. (2006). Demographic and Background Questionnaire. (CREATE Technical Rep. CREATE-2006-02).

Elsmore, T. F. (1994). SYNWORK1: A PC-based tool for assessment of performance in a simulated work environment. *Behavior Research Methods, Instrumentation, and Computers, 26,* 421-426.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*(1), 32-64.

Endsley, M. R. & Kiris, E. O. (1995). The out-of-the-loop performance problem and level of control in automation. *Human Factors, 37*(2), 381-394.

Fisk, A. D. & Rogers, W. A. (1991). Toward an understanding of age-related memory and visual search effects. *Journal of Experimental Psychology: General, 120*(2), 131-149.

Herrnstein, R. J. (1961). Relative and absolute strength of response as a function of frequency of reinforcement. *Journal of the Experimental Analysis of Behavior, 4,* 267-272.

Jones, D. G. & Kaber, D. B. (2005). Situation awareness measurement and the situation awareness global assessment technique. In N. A. Stanton (Ed.), *Handbook of Human Factors and Ergonomics Methods* (pp. 42-1-42-7). Boca Raton, FL: CRC Press.

Ma, R. & Kaber, D. B. (2007). Situation awareness and driving performance in a simulated navigation task. *Ergonomics, 50*(8), 1351-1364.

McBride, S.E., Rogers, W.A., & Fisk, A.D. (2010). Using an Automated System: Do Younger and Older Adults Differentially Depend? *Proceedings of the Human Factors and Ergonomic Society 54th Annual Meeting.* San Francisco, CA: Human Factors and Ergonomics Society.

McClelland, J. L. (1979). On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86*(4), 287-330.

Navon, D. & Gopher, D. (1979). On the economy of the human-processing system. *Psychological Review, 86*(3), 214-255.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, 30*(3), 286-297.

Schneider, W. & Fisk, A. D. (1982). Concurrent processing and controlled visual search: Can processing occur without resource cost? *Journal of Experimental Psychology: Learning Memory and Cognition, 8*(4), 261-278.

Schneider, W. & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. detection, search, and attention. *Psychological Review, 84*(1), 1-66.

Shipley, W. C. (1986). *Shipley institute of living scale*. Los Angeles: Western Psychological Services.

Sit, R. A. & Fisk, A. D. (1999). Age-related performance in a multiple-task environment. *Human Factors, 41*(1), 26-34.

Snellen, H. (1868). Test-types for the determination of the acuteness of vision (4th ed.). London: Williams & Norgate.

Sternberg, S. (1969). Memory-scanning: Mental processes revealed by reaction-time experiments. *American Scientist, 57*(4), 421-457.

Wechsler, D. (1997). *Wechsler adult intelligence scale III. (3rd Ed.)*. San Antonio, TX: The Psychological Corporation.

Wickens, C. D. (1980). The structure of attentional resources. In R. Nickerson's (Ed.) *Attention and Performance VIII*. Hillsdale, NJ: Erlbaum.

Wickens, C. D. (1984). Processing resources in attention. In R. Parasuraman and D. R. Davies (Eds.), *Varieties of Attention* (pp. 63-101). New York: Academic Press.

Wickens. C. D. & Dixon, S. R. (2007). The benefits of imperfect diagnostic automation: A synthesis of the literature. *Theoretical Issues in Ergonomics Science, 8*(3), 201-212.

Wickens, C. D. & McCarley, J. S. (2008). *Applied Attention Theory.* Boca Raton, FL: CRC Press.